

Dissecting the Software Patent Problem

An Argument Against Patentability based on the
Relationship between Software and Mathematics

Anton Robert Hughes

BE (CSE), LLB (Hons), Grad Dip Legal Prac, Grad Cert Commercialisation

Submitted in fulfilment of the requirements for
the Degree of Doctor of Philosophy

May 2012



University of Tasmania

Declaration of Originality

This thesis contains no material which has been accepted for a degree or diploma by the University or any other institution, except by way of background information and duly acknowledged in the thesis, and to the best of my knowledge and belief no material previously published or written by another person except where due acknowledgment is made in the text of the thesis, nor does the thesis contain any material which infringes copyright.

Authority of Access

This thesis may be made available for loan and limited copying and communication in accordance with the Copyright Act 1968.

Statement of Ethical Conduct

The research associated with this thesis abides by the international and Australian codes on human and animal experimentation, the guidelines by the Australian Government's Office of the Gene Technology Regulator and the rulings of the Safety, Ethics and Institutional Biosafety Committees of the University.

Anton Robert Hughes _____ Date _____

For Bob

Abstract

Should software be patentable? Despite a US Presidential Commission answering in the negative in 1968, and a legislative exclusion operating in the UK since 1977, the patenting of software has become a regular occurrence in the US, the UK and Australia. But even now software's patentability is not settled, as evidenced by the widespread protests against the EU Software Directive in 2005, and the level of anticipation of the US Supreme Court decision in *Bilski v Kappos* in 2010. The reason for this continuing unrest is that the patenting of software provides a number of practical problems for the software industry and theoretical challenges to the coherence of the patent regime.

In all of this, software's relationship with mathematics has been overlooked. Software is both a product of and isomorphic to mathematics. This makes for an interesting avenue of inquiry, both because of mathematics' long history, and courts' acceptance of it as inherently non-patentable. This thesis explores historical and philosophical accounts of mathematics in pursuit of a better understanding of its nature. That account demonstrates why many theories as to mathematics' non-patentability are largely unsatisfactory. However, by refocusing the debate on the conditions necessary for mathematical advancement, a three-dimensional analytical framework emerges, centred around the concept of the useful arts. This analysis both explains mathematics' non-patentability, and offers a theory of patentable subject matter consistent with Australian, American and European patent law. The analysis is then applied to the field of software to explain why software falls so close to the boundary, but ultimately ought not be considered patentable subject matter.

Contents

Abstract	i
Contents	iii
Acknowledgments	ix
Introduction	1
1 Aims	5
2 Outline	7
3 Other matters	9
1 The Nature of Software	11
1 Introduction	11
2 What is software?	11
The Evolution of Modern Software	12
Machine Code	12
Assembly Language	13
High level languages	13
Compilers	14
Interpreters	14
Firmware, kernels and operating systems	14
Libraries, frameworks and patterns	15
Scripting languages and domain-specific languages	16
The modern software development process	17
Main purpose	17
System architecture	17
Abstract data types (ADTs)	18
Algorithms and Data Structures	18
Source code	20
Object code	20

Contents

	Putting it all together	22
3	An example	24
4	How is software related to mathematics?	26
	Formal equivalence: the Turing machine	28
	Are computers and the Turing machine equivalent?	31
	Criticisms of the Church-Turing thesis	32
	Structural similarities between mathematics and software	33
5	Implications of the isomorphism for patent law	35
6	Conclusion	36
2	The History of Patentable Subject Matter	37
1	Introduction	37
	What is a patent?	38
	What is a manner of manufacture?	39
2	Early history	40
3	Post-1623 developments	42
	Boulton v Bull	42
	Principles	43
	Processes	43
	Overall	44
	After Boulton v Bull	45
	Patents as a social contract	45
	Towards an American Patent System	46
	Analogous uses	47
	Early US case law	48
	An Australian patent custom?	49
4	The modern patent system emerges in the UK	49
5	The early 20 th Century	51
	United Kingdom	51
	Non-patentability of principles revisited	52
	Commerce and trade	56
	Morton's rules	56
	United States	59
	Australia	59
6	The NRDC case	61
	Analogous uses	62
	Inventiveness; discovery v invention; knowledge v application	63
	Scope of patentable subject matter	64
	Application to the present case	69
7	Conclusion	71
3	How Software Became Patentable	73
1	Introduction	73

Contents

2	The emergence of the modern computer	73
3	How software became patentable subject matter	77
	The United States	77
	Mental steps doctrine	78
	Gottschalk v Benson	86
	Parker v Flook	88
	In re Freeman & In re Walter	89
	Diamond v Chakrabarty	90
	Diamond v Diehr	91
	In re Abele	93
	Alappat	93
	State Street	94
	Ex parte Lundgren	96
	The United Kingdom	98
	Developments under the Patents Act 1949 (UK)	99
	The European Patent Convention	102
	Vicom	105
	Genentech Inc's Application	107
	Merrill Lynch	109
	Gale's Application	109
	Fujitsu Ltd's Application	111
	The IBM Twins	114
	The Board Diverges	115
	Software patent directive	117
	Australia	118
	Early Patent Office practice	118
	IBM v Commissioner of Patents	121
	CCOM Pty Ltd v Jiejing Pty Ltd	123
	Welcome Real-time SA v Catuity Inc	125
4	Recent Developments	127
	Australia	128
	United Kingdom & European Union	131
	Aerotel/Macrossan	131
	Symbian	136
	Referral to the Enlarged Board (G3/08)	138
	The United States	141
	In re Comiskey	141
	In re Bilski	141
	Bilski v Kappos	142
	Overview of recent developments	147
5	Conclusion	147
4	Why Software Patents are a Problem	149

Contents

1	What is a software patent?	149
2	Why is the patenting of software problematic?	150
	Abstraction	151
	<i>Pre-emption/breadth</i>	153
	<i>Mental steps doctrine</i>	154
	<i>Intellectual information exclusion</i>	155
	Physicality	161
	Mischaracterisation	164
	Complexity	172
	Reuse	174
	Software innovations are largely cumulative	174
	The software industry relies on openness	174
	Summary	177
3	How does the award of patents affect the software industry?	177
	Poor documentation means more trivial patents	177
	The software market moves too fast for the patent regime	178
	Network effects amplify the power of patent holders	179
	Programmers hate software patents	179
4	Is this a subject matter issue?	181
5	How should the analysis proceed?	183
6	Conclusion	184
5	The Nature of Mathematics	187
1	A Mathematicians' Account of Mathematics	188
	The history and philosophies of mathematics	193
	Ancient times	194
	15th Century	196
	17th Century	197
	19th Century	200
	20th Century	206
	Can these various views be reconciled?	211
2	A Patent Lawyer's Account of Mathematics	213
	Europe	214
	European Patent Office	214
	United Kingdom	216
	The US	218
	Early jurisprudence	219
	Yuan	219
	Benson and Diehr	220
	Alappat and State Street	220
	Pre-Bilski practice	222
	Bilski v Kappos	223
	Australia	224

	Why is mathematics non-patentable?	227
3	Conclusion	230
6	Why Mathematics Is Not Patentable	233
1	Introduction	233
2	Freedom in patent law	234
3	What is meant by freedom?	236
	Freedom of Thought	236
	Freedom of Expression	238
	<i>Truth</i>	239
	<i>Autonomy</i>	240
	<i>Democracy</i>	241
4	Is freedom required in mathematics?	242
	Thought in mathematics	243
	Expression in mathematics	247
	Expression in the philosophies of mathematics	247
	Mathematics as a language	249
	Mathematics as an aesthetic activity	252
5	Freedom, the useful arts and non-patentability	255
	Defining the fine arts and useful arts	255
	What are the fine arts?	256
	What are the useful arts?	258
	The three dimensions of the fine arts versus useful arts distinction	268
	Aesthetic versus rational	269
	Expressive versus purposive	270
	Intangible versus physical	270
	Summary	270
6	Conclusion	271
7	Why Programming is not among the Useful Arts	273
1	Introduction	273
2	Is programming a fine art or a useful art?	274
	Intangible or physical?	275
	The externalisation of thought	277
	Summary	281
	Expressive or purposive?	281
	Summary	291
	Aesthetic or Rational?	292
	Just another type of engineering?	293
	No Silver Bullet	295
	Evaluative aesthetic	297
	Motivational aesthetic	298
	Generative aesthetic	302

Contents

Summary	306
Programming is not in the useful arts	306
3 Conclusion	308
8 Implications	309
1 Introduction	309
2 An assessment of the framework	310
3 An assessment of the state of patent law	310
The United States	311
The United Kingdom	312
Australia	313
Summary of the jurisdictions	314
4 Broader implications of the argument for patent law	315
Subject matter is not a “failed gatekeeper”	315
Alternative solutions are insufficient	315
Giving effect to the purpose of patent law	319
Characterisation is key	322
Generalisations <i>and</i> case-by-case analysis are required	323
Human rights jurisprudence	324
5 Beyond the patent paradigm	327
Trade secrets	328
Copyright	330
Between the cracks?	331
An alternative protection paradigm?	331
Conclusion	335
Bibliography	339
Articles/Books/Reports	339
Cases	366
Legislation	373
Treaties	374
Other	374

Acknowledgments

First and foremost, I have to thank my wife Sarah, who has physically, emotionally and intellectually supported me through this entire process. Aside from forbidding the discussion of mathematics after 10pm, Sarah has listened attentively on myriad occasions as I voiced barely comprehensible diatribes on topics whose relevance is peculiar to me alone. Thank you also for digging deep in the final run to the finish, and proof reading with the sort of dedication I have come to expect from you.

I am also greatly indebted to my supervisor, Professor Dianne Nicol for her selfless dedication and support. Despite resolute denials as to expertise in the areas of mathematics and software, Di has at every step of the way provided insightful and encouraging feedback which motivated me to work harder, and reach higher.

I would also like to thank the Faculty of Law for extending me a scholarship, and giving me the resources I need to get to work. I have always been afforded generous and unquestioning support by the past and present Deans, Don Chalmers and Margaret Otlowski, and the staff and postgraduates of the Faculty have always made me feel welcome.

A sincere thankyou must also go to the Honourable Justice Porter, who afforded me the opportunity of a front row seat for 2 years to watch the law unfold. Thanks to your your guidance and support, I learned much about thinking in issues, attention to detail and the value of a blank piece of A3 paper.

Thank you to my mother Desma, who understood from personal experience that the process of writing a PhD could be difficult, and whose advice about the process of finishing helped me greatly in the final stages. Thanks for all that hard work proof reading at short notice, and the positive encouragement that went with it. Thank you to my sister Natalya for the advice about aesthetics and semiotics – we truly had the coolest conversations ever.

A special thank you to Alice Sorcha and Niamh Milly, who only arrived after this project started, but have made my days and nights delightful ever since. Thanks for your support.

And finally, to my father Bob, who inspired me to want to start a thesis even before I really knew what one was. 25 years later that project is complete. Thanks for the many words of wisdom that, although said almost in passing, somehow resound at just the right moments.

Introduction

In modern life, software has become ubiquitous. The primary vehicle by which has software entered the public consciousness is the personal computer. It is not unusual for people to have a computer at work, and a computer at home. But people interact with and even depend on software in increasingly varied ways. Software runs on all manner of devices including “clothes washers and dryers; toasters and microwave ovens; electronic organisers; digital televisions and digital audio recorders; home alarm systems and elderly medic alert systems; irrigation systems; pacemakers; video games; Web-surfing devices; copying machines; calculators; tooth-brushes; musical greeting cards; pet identification tags; and toys.”¹ The modern mobile phone is a pocket computer,² and its software forms a key point of differentiation for many buyers.³ Cars are increasingly reliant on software components, embedded in small microcontrollers, to manage emissions and fuel economy; for advanced diagnostics; to simplify manufacture and design; as well as for comfort and convenience features.⁴ Complex software applications drive cutting-edge scientific research projects, for example the Large Hadron Collider,⁵ and Human Genome Project.⁶ Software systems are used to manage and control the delivery of infrastruc-

¹ Martin L Shooman, *Reliability of computer systems and networks: fault tolerance, analysis and design* (John Wiley and Sons, 2002) at 9.

² The Android operating system, which powers a growing number of smart phones, is made up of approximately 12,000,000 lines of code: Angel Leon, “How many lines of code does it take to create the Android OS?” on *Gubatron.com* <<http://www.gubatron.com/blog/2010/05/23/how-many-lines-of-code-does-it-take-to-create-the-android-os/>> (4 September 2011).

³ Buyers are increasingly interested in whether their phones run on iOS, Android, Windows, Blackberry OS or Symbian. See for example Denis Gallagher, “Smartphone operating systems review” *Choice Online*, 30 August 2011 <<http://bit.ly/qk646R>> (14 September 2011).

⁴ Karim Nice, “How Car Computers Work” on *HowStuffWorks.com* <<http://auto.howstuffworks.com/under-the-hood/trends-innovations/car-computer.htm>> (4 September 2011).

⁵ For an overview, see Wikipedia, “Large Hadron Collider” <http://en.wikipedia.org/wiki/Large_Hadron_Collider> (4 September 2011). The Large Hadron Collider depends software to control the operation of the multitude of hardware components, as well as to collect and analyse data. For an overview of the software involved, see Juan Batiz-Benet, Xuwen Cao and Yin Yin Wu, “CERN’s Large Hadron Collider” <<https://sites.google.com/site/multinationalsoftwares/multinationalprojects/cern-lhc>> (4 September 2011).

⁶ The goals of the HGP were to “identify all the approximately 20,000-25,000 genes in human DNA; determine the sequences of the 3 billion chemical base pairs that make up human DNA; store this information in databases; improve tools for data analysis; transfer related technologies to the private sector; and address the ethical, legal, and social issues (ELSI) that may arise from the project”: United States Department of Energy, “About the Human Genome Project” <http://www.ornl.gov/sci/techresources/Human_Genome/about.shtml> (4

Introduction

ture services such as electricity, water, telecommunications and public transport. In addition to the wide range of contexts in which software is used, the size and complexity of individual programs has grown exponentially.⁷ In 1981, the MS-DOS operating system which powered most personal computers, was comprised of around 4000 lines of code.⁸ By 2002, its ancestor, Windows XP, contained in the region of 40,000,000 lines of code.⁹

As a result, the software industry is big business. According to the US Bureau of Labor Statistics, in 2008 there were around 1.3 million computer systems software engineers and programmers in the US alone. That number was projected to increase by 21%, to around 1.6 million, by 2018.¹⁰ The size of the global software industry has been estimated to be worth US\$225.5 billion in 2009. By 2014, that figure is expected to increase by 32.6% to a figure of around US\$299 billion.¹¹

But the software business is not like other businesses. Much of the reason for that stems from the unusual nature of the product. Software is “infinitely malleable”.¹² It is an arrangement of concepts, constructed in words. Because of this, it is not engineered like traditional physical products. It does not progress sequentially through various phases from design, to prototype, to testing, to production, with the laws of physics throwing up unexpected difficulties at each stage along the way. Unlike a car, any part of a software product can be completely changed at any point along the development pipeline, and updates continue to be released after the product is distributed. No matter how much cost is involved in producing the first copy, subsequent copies can be made for free. Software might be built for one customer, or a million: the development process is the same. Software is regularly built to order, bought off the shelf, or a hybrid of the two – an off-the-shelf product selected, then customised for the specific requirements of an individual or an organisation.

The flexibility and freedom of software is both its strength and its weakness. The ability to modify a product at any point, even after it has been released on the open market, offers users the potential to adapt software to their needs, rather than adapting their needs to the software. But the temptation to change can be hard to resist. This can make project requirements a con-

September 2011). The project was completed in 2003, although the data analysis continues. On the role of automation in the project, see Lynn Yaris, “Machines and the Human Genome Project”

<www.lbl.gov/Science-Articles/Archive/human-genome-mapping-sequencing.html> (4 September 2011):

“‘From the beginning, laboratory automation has been recognized as an essential element of the Human Genome Project,’ says Ed Theil a computer systems engineer who has been working with the Human Genome Center’s instrumentation group. ‘Beyond the advantages of speed and relief from tedium, automation minimizes human errors and captures data instantly as part of the process.’”

⁷ On the relationship between size and complexity, see Chapter 1, at 23.

⁸ Henry M Walker, *The tao of computing* (Jones and Bartlett Learning, 2004) at 79.

⁹ Henry M Walker, *The tao of computing* (Jones and Bartlett Learning, 2004) at 79. It is acknowledged of course that the number of lines of code in an application is a very crude measure, as it does not take account of language choice, or programming style, both of which may produce marked variations in lines of code used for similar tasks.

¹⁰ United States Bureau of Labor Statistics, “Computer Software Engineers and Computer Programmers” in *Occupational Outlook Handbook, 2010-11 Edition* <<http://www.bls.gov/oco/ocos303.htm>> (3 September 2011).

¹¹ DataMonitor, “Software: Global Industry Guide 2010” 31 January 2011

<http://www.datamonitor.com/store/Product/software_global_industry_guide_2010?productid=4F026C5C-EBCC-4193-AD27-77260196E7F5> (3 September 2011). DataMonitor did note however a shrink in the market from 2008 of 5.4%. For further statistics Business Software Alliance, “Software Industry Facts and Figures” <<http://bit.ly/nq8jKP>> (3 September 2011).

¹² Frederick P Brooks Jr, “No Silver Bullet: Essence and Accidents of Software Engineering” (1987) 20(4) *Computer* 10 at 12.

stantly moving target. As a result, software projects are notorious for blown budgets and late delivery.

The software “factory” is also unusual. Despite the size and complexity of software creations, they can be built by comparatively small teams.¹³ Those teams can share an office, or be scattered around the globe. Members of the team, the programmers, might see themselves as software engineers, but they may well consider themselves artists. Given the intangibility of the medium they work with, it can be hard to measure their productivity. Empirical evidence suggests a ten-fold variation, even amongst people with similar experience.

In light of these many unique features, it seems sensible to question whether software should be patentable. Traditionally, the patent system has been an important mechanism for benefiting society through the encouragement of innovation. Patent law is said to achieve such encouragement in three ways: the award of monopolies which encourages both the creation of new inventions, and their commercialisation; through the disclosure of those technologies to the public, which encourages innovation in related areas; and through.¹⁴

But patent law came into being early in the 15th Century, and took its modern form towards the end of the end of the 19th Century. Its original purpose was to incentivise the establishment of new industries at the start of the Industrial Revolution,¹⁵ a time where inventing and commercialising new products was a slow process of trial and error. Patent law protected the inventor’s good idea against the open market to allow it to be commercialised. Since that time, advances in science and technology have seen the patent system broaden to cover a new breed of less and less tangible invention, from chemical processes, to methods of improving devices, methods of medical treatment and even the process of isolating biological materials.

Software is a highly intangible and complex creation, and its industry is characterised by a rapid produce lifecycle, and a high degree of dependence on reuse and conformity. Thus when the question of software’s patentability first fell to be considered in the 1960s, it should be considered a bridge too far for the patent regime. The early indications were that it was. In 1966 a US Presidential Commission composed of leading academics, industry representatives and the Commissioner of Patents recommended that Congress legislatively exclude software from patent law.¹⁶ In Europe, just such an exclusion of computer programs forms part of the European Patent Convention,¹⁷ and has been given legislative operation in the UK since 1977.¹⁸ In Australia, early decisions of the Australian Patent Office were clearly against software patents, although the patentability of software was not judicially considered until 1991.¹⁹

¹³ “I’ve been heard word processors seriously compared with the complexity of Boeing’s airplanes. But according to the product manager of Microsoft Word (version 3.1), his software development team consisted of only eight programmers.”: Brad Cox, “No Silver Bullet Revisited” *American Programmer Journal*, November 1995 < <http://virtualschool.edu/cox/pub/NoSilverBulletRevisited> > (22 July 2011)

¹⁴ Kathy Bowrey, Michael Handler and Dianne Nicol, *Australian Intellectual Property: Commentary, Law and Practice* (Oxford University Press, 2010) at 378.

¹⁵ A classic example of this Industrial Era invention is Watt’s steam engine, an improvement of which was the subject of *Boulton v Bull* (1795) 126 ER 651. The case is discussed in Chapter 2 at 42.

¹⁶ Presidential Commission on the Patent System, *To Promote the Progress of Useful Arts in an Age of Exploding Technology* (1966).

¹⁷ *Convention on the Grant of European Patents*, opened for signature 5 October 1973, 13 ILM 268 (entered into force 7 October 1973).

¹⁸ *Patents Act 1977* (UK).

¹⁹ *IBM v Commissioner of Patents* (1991) 33 FCR 218.

Introduction

Despite this early reticence, the patenting of software has become commonplace in the US, the UK and Australia. But this does not mean that software's patentability is a settled issue. The path towards software's patentability was a gradual one, which was arguably not completed in the US until the decision in *State Street*²⁰ in 1998. But the campaign against software patenting remained vociferous, and determined. The widespread protests against the EU Software Directive in 2005, seen to allow for the uninhibited patenting of software in that jurisdiction, eventually won the day, and collected nearly half a million signatures along the way.²¹ Even now, outraged members of the blogosphere, including many software developers, rail against what they see as a broken patent system, and the ludicrousness of software patenting. The recent outbreak of the smartphone software patent wars, pitting iPhones against Androids, have added to the discontent.²²

So despite the passage of over 40 years since the patentability of software was first considered, the question of software's patentability remains uncertain. Misunderstandings as to software's nature, based on intuition rather than analysis, have combined with a preparedness to expand traditional understandings of inherent patentability to accommodate it. This has led to practical problems for the software industry, where patents are used less as an aid to innovation, and more as a defensive mechanism. This accommodating attitude also presents theoretical challenges to the coherence of the patent regime. On one level, software is an intellectual creation of the kind more familiar to the copyright regime. On another level, software controls the operation of a functional device – a computer – and such a machine seems a comfortable fit within patent law's historical protection of all manner of useful devices. The problem, put simply, is that software threatens to stretch the boundaries of patentable subject matter into uncharted territory – the realm of the abstract, intellectual creation. This may be the coal-face of innovation in an Information Age, but whether patent law's Industrial Era regime provides an appropriate incentive is far from clear.

In all of this, software's relationship with mathematics has been overlooked. Software is a by-product of a quest for truth in mathematics, having developed out of symbolic logic, and mathematical formalism. Symbolic logic seeks to reduce verifiability of mathematical results to the mechanical application of a sequence of rules, starting from agreed axioms. Mathematics and software are also isomorphic activities, in that the activities involved in doing mathematics, and the activities involved in developing software closely correspond to each other. This point is related to the last, in that both disciplines seek to reduce knowledge to a logical sequence of steps – one for the purpose of verification, the other for instructing a computer. This close

²⁰ *State Street Bank & Trust Co v Signature Financial Group Inc* 149 F.3d 1368 (Fed Cir, 1998).

²¹ A repeat performance seems likely, over the proposed introduction of a unitary patent system in the EU. See the discussion below.

²² See for example Chloe Albanesius, "Infographic: Who Will Win the Patent Wars?" *PCMag.com*, 2 September 2011 <<http://www.pcmag.com/article2/0,2817,2392375,00.asp>> (4 September 2011); L Gordon Crovitz, "Google, Motorola and the Patent Wars" *The Wall Street Journal*, 22 August 2011 <<http://online.wsj.com/article/SB10001424053111903639404576518493092643006.html>> (4 September 2011); Holman W Jenkins, "Obama and the Smartphone Wars" *The Wall Street Journal*, 24 August 2011 <<http://online.wsj.com/article/SB10001424053111903327904576526130093390612.html>> (4 September 2011); Kimberlee Weatherall, "Samsung Galaxy Tab vs Apple iPad: the tablet patent wars hit Australia" *The Conversation*, 3 August 2011 <<http://theconversation.edu.au/samsung-galaxy-tab-vs-apple-ipad-the-tablet-patent-wars-hit-australia-2660>> (4 September 2011).

relationship makes the status of mathematics in patent law an interesting avenue of inquiry. Unlike software, mathematics has a history arguably as long as humanity itself. So considering the patentability of mathematics, which seems always to have been non-patentable, offers a rich repository from which to draw in deciding this contemporary issue.²³

1 Aims

Briefly stated, the aims of this thesis are as follows:

1. To argue that software ought not be patentable.
2. In making that argument, to incorporate considerations that go beyond a utilitarian, or purely economic, analysis.
3. To shed light on the nature of software, and the way in which it is constructed, in order to better inform debate, both in the context of intellectual property, and in other legal contexts upon which such an understanding depends.

This thesis is primarily an argument against the patentability of software. It will be shown that the nature of software, and perhaps more importantly, the way it is created, makes it of a kind which cannot be reconciled with the history, purpose and operation of the patent regime. But it is not the first argument advanced for that purpose, nor is it likely to be the last.

Therefore, it is intended that the argument be differentiated from others not by the ends, but by the means. Boyle notes that much of the critical IP scholarship takes place in a “relentlessly utilitarian framework”.²⁴ As such, patents are often justified on purely economic grounds. Patent theory asserts that “where the norm of free competition would result in free riding by competitors and less reason to invest in new technologies, innovation is encouraged by providing innovators with the exclusive rights to their inventions.”²⁵ Innovation is considered to be a positive on the basis that the public benefit from the disclosure of new and useful technologies, which fall into the public domain at the expiry of the patent grant period – the so-called social contract theory of patent law.²⁶ These benefits must be contrasted with the award of monopoly

²³ “[A] page of history is worth a volume of logic”: *New York Trust Co v Eisner* 256 US 345 (1921) per Holmes J; “In this bright future, you can’t forget your past.”: Bob Marley, “No Woman, No Cry”, *Live!* (1975).

²⁴ James Boyle, “Enclosing the Genome: What the Squabbles over Genetic Patents Could Teach Us” in F Scott Kieff, *Perspectives on the Human Genome Project* (Academic Press, 2003) 97 at 109.

²⁵ Advisory Council on Intellectual Property, *Patentable Subject Matter: Issues Paper* (2008) <<http://www.acip.gov.au/library/Patentable%20Subject%20Matter%20Issues%20Paper.pdf>> (accessed 13 November 2008) at 1.

²⁶ This line of thinking emerged *Liardet v Johnson* (1778) 1 Carp Pat Cas 35 (NP), discussed in Chapter 2 at 45. See also *Turner v Winter* (1787) 19 Eng Rep 1276 where the Court noted that “[t]he consideration, which the patentee gives for his monopoly, is the benefit which the public are to derive from his invention after his patent is expired” However, economic rationales for patent law began to dominate as the eighteenth century progressed. See Edward C Walterscheid “The Early Evolution of the United States Patent Law: Antecedents (Part 4)” (1996) 78 *Journal of the Patent and Trademark Office Society* 77 at 104-106. Similarly see the second reading speech for the *Patents Act 1990* Australia: “[t]he essence of the patent system is to encourage entrepreneurs to develop and commercialise new technology.”: Commonwealth of Australia, “Patents Bill 1990: Second Reading” Senate, 29 May, 1990 <http://parlinfoweb.aph.gov.au/PIWeb/view_document.aspx?id=562046&table=HANSARDS> (2 November 2004). A different theory underlies the US system, namely patents as a reward of inventors: Edward C Walterscheid “Patents and Manufacturing in the Early Republic” (1998) 80 *Journal of the Patent and Trademark Office Society* 855 at 856.

Introduction

rights, which amount to a tax on society

in two different ways: first, by the high price of goods, which, in the case of a free trade, they could buy much cheaper; and, secondly, by their total exclusion from a branch of business which it might be both convenient and profitable for many of them to carry on.²⁷

A narrow view of social contract theory frames the patentable subject matter question as a question of whether the economic benefits of innovation, prevention of free-riding, and incentives to invest outweigh the increased prices and exclusion of competition. The danger of assessing the cost on a purely economic basis in these circumstances is that the costs of patenting might be social, ethical, religious, environmental or scientific, and do not fit neatly into an economic analysis. The dangers of overstating patent law in purely economic terms can be stated thus:

Economics can be useful when it is viewed as a science which examines the decision making process, the study of optimization behaviours subject to constraints. Economics cannot be helpful if it is viewed as a precise tool that can mechanically and independently determine the outcomes of complex problems.... [I]t can be used to assist in the framing of issues and in isolating the appropriate factors for judicial consideration. Economics is not helpful, however, in the inherently subjective process of weighing and quantifying competing concerns. It is wrong not to recognize this limitation, and it is dangerous to assume that difficult, value-laden decisions ... can be decided mechanically by appealing to an economic formula.²⁸

Such a question properly recognises that at times questions of patentability cover social, ethical, religious, environmental, and scientific issues, in addition to economic and legal ones. The latter by themselves will sometimes form an insufficient basis on which to ground patent law. What is sought to be achieved in the course of this argument then, is not to displace economics as a tool by which to assess patent law. It is to develop an alternative analytical framework by which to assess patentable subject matter, and one which at the very least leaves the door open to these broader considerations. At the same time, it is recognised that any such analysis needs to be consistent with the principles developed over patent law's very long history.

The final aim of this thesis is to bring some clarity to the nature of software. The nature of software as both a description of a computable process, and the mechanism by which such a process is actually performed, is apt to cause confusion. For example, it may be difficult for some to understand how something which can be bought in a box at a shop, carried home, inserted into a computer, and then run, causing certain images to appear on a screen, might not in fact be a physical artefact at all. By applying the analytical framework which will be developed in this thesis to the special case of software, it is hoped that some light can be shed on the nature of both software development, and software itself, and avoid further mischaracterisation in the future.

²⁷ Adam Smith, *An Inquiry into the Nature and Causes of the Wealth of Nations* (5th ed, Methuen, 1904) at 159-160.

²⁸ Peter J Hammer, "Free Speech and the 'Acid Bath': An Evaluation and Critique of Judge Richard Posner's Economic Interpretation of the First Amendment" (1988) 87(2) *Michigan Law Review* 499 at 499.

2 Outline

Chapter 1 explores important introductory material, namely the nature of software, although many of the ideas explored are developed in more detail in Chapter 7. In particular, three aspects of software are identified as key to a proper understanding of it. Firstly, successive generations of software have been freed from specific hardware limitations through a process of abstraction. Understanding the level of abstraction means understanding the connection between the alleged invention and the physical constraints of the computer. The level of abstractness varies depending on the programming task at hand, and therefore analysis can be very context-specific. Secondly, software development depends on reuse, with software components being built upon, or with, layers, libraries, frameworks and design patterns. Reuse is a key characteristic of the software industry, and one without which it could not continue. Finally, software entities are highly complex, typically involving large numbers of unique components. The scale of complexity has been said to be similar in magnitude to that of an aircraft, but the unique nature of each component means there are no economies of scale. Chapter 1 also explores the nature of the relationship between software and mathematics, and explain how it is that software is said to be identical with, or at least isomorphic to, mathematics at both formal structural levels.

Chapter 2 embarks upon an historical account of the patentable subject matter inquiry, tracing the development of such principles from the earliest known patent customs in Europe to the most recent developments in three jurisdictions – the United Kingdom, the United States of America, and Australia. It will be shown that a number of analytical tools have been used to help define the borderline between what is patentable and what is not. A key theme however has been the contrast between abstract, intangible subject matter which is not patentable, and those subject matters involving a corporeal, tangible manifestation which are. The chapter concludes with an analysis of the leading Australian case on patentable subject matter, *National Research Development Corporation v Commissioner of Patents*²⁹ The case has been described as a watershed³⁰ and is sometimes characterised as a break with the past.³¹ However, it will be argued that rather than representing a new phase in the development of patent law, *NRDC* was intended as a continuation of existing principles, and that any claim to a “formula” arising from the case is a misapplication of its text.

Having introduced the key features of computer software and the historical development of the patentable subject matter inquiry it becomes possible to draw those two strands together. Chapter 3 first considers the early history of modern computing, and the emergence of software as a separate phenomenon to the hardware upon which it runs, before charting the course of software from non-patentable idea (or mental process) to patentable process in all three jurisdictions, despite a clear exclusion in the UK. The Chapter concludes by analysing the current

²⁹ (1959) 102 CLR 252 (“NRDC”).

³⁰ *Joos v Commissioner of Patents* (1972) 126 CLR 611 at 616 per Barwick CJ. See also *CCOM Pty Ltd & Anor v Jiejing Pty Ltd & Ors* (1996) 21 FCR 260 at 287.

³¹ See for instance William van Caenegem, “The Technicality Requirement, Patent Scope and Patentable Subject Matter in Australia” (2002) 13 *Australian Intellectual Property Journal* 41; *CCOM Pty Ltd & Anor v Jiejing Pty Ltd & Ors* (1996) 21 FCR 260 at 287: “[NRDC] changed the direction of case law not only in Australia but also in the United Kingdom.”

Introduction

state of the patentable subject matter inquiry, both as it relates to software, and more broadly for other abstract subject matter.

Chapter 4 will explain in detail why the patenting of software is problematic from both practical and theoretical perspectives, tracing these difficulties to the three key features of software identified in Chapter 1. The chapter will also document the way in which the courts of the US, the EU and Australia have attempted to deal with these problems, and describe why it is that the software patent issue should be dealt with as a patentable subject matter issue, rather than as an issue of novelty or non-obviousness. It will be shown why the current approaches are insufficient, and to introduce the notion that a fresh approach to the software patent problem is needed.

The ultimate aim of the thesis is to shed fresh light on the scope of patentable subject matter, to clarify the nature of software, and by doing so to explain why software should not be patentable subject matter. Chapter 5 begins that journey by building on the isomorphism between software and mathematics identified in Chapter 1. The chapter firstly explores the history and philosophies of mathematics in an attempt to ascertain its nature. It will then be shown that legal accounts of the nature of mathematics are insufficient, as they cannot be reconciled with the contributions of the various philosophies of mathematics to a full understanding of what the nature of mathematics is.

Chapter 6 further explores the nature of mathematics, with a view to constructing a more suitable explanation of why mathematics is non-patentable. To do so, a different approach is taken. Whereas the historical, philosophical and legal accounts of the nature of mathematics have focused on what mathematics *is*, it will be shown that the better focus is on what mathematics *requires* for its further progress. By answering that question, it will be possible to avoid the impossibility of providing a definitive unifying account of what mathematics is, and to answer the question of whether mathematics is a proper subject for patent protection. It will be shown that the underlying reason for excepting mathematics from patentability is that mathematical innovation depends on freedom of thought and expression.

Rather than seeking to recharacterise patent law as the competition between inconsistent human rights however, it will be shown how such freedoms can be reconciled with a traditional understanding of the scope of patent law. The mechanism by which that reconciliation is to be achieved is the distinction between the non-patentable fine arts and the patentable useful arts. It will be argued that the concept of the useful arts, or technology, as it might be known in contemporary language, provides a suitable way of distinguishing between invention and other intellectual pursuits. The analysis of the way freedom operates with mathematics, combined with a foray into the philosophy of technology, leads to the derivation of a three-dimensional analytical framework by which it is possible to distinguish between the fine arts and the useful arts in a structured fashion. The analysis is then applied to mathematics to show why mathematics is a fine art and not a useful art, notwithstanding its role in many technological pursuits, because it is intangible rather than physical, expressive rather than functional, and aesthetic rather than rational.

Chapter 7 applies the three dimensional framework to determine how far the isomorphism

and formal connection extend the asserted basis of the non-patentability of mathematics to software. The expressive, aesthetic and intangible nature of software development will be explored. The existence of these attributes, it will be argued, supports the contention that the classification of mathematics as a fine art will usually be appropriate to software. In particular, it will be suggested that the interrelationship of software and hardware, referred to as abstractness in Chapter 1, acts as an important consideration. Where the software and hardware are closely interrelated, the physical limitations may constrain the software such that expressive and aesthetic considerations are similarly limited. Where software is merely ancillary to a physical device, this does not change the nature of programming, but it may change what is sought to be patented, namely a physical device in which software is merely a subsidiary component. The limited scope of the claims will mean both that programming as a creative activity is not inhibited, and that what is claimed is sufficiently connected to the traditional conception of the useful arts that it is not contentious. Determining whether it is software, or a physical device which is actually being claimed requires difficult determinations of fact in any particular case, but the difficulty of the choice does not obviate the need to draw such distinctions.

Having demonstrated why programming, like mathematics, is not a useful art, the final chapter, Chapter 8, explores the implications of that position. The value of the framework, and the current state of patent law in Australia, the UK and the US are assessed. The importance of characterisation is also acknowledged. Next, the role of subject matter is considered. The need to exclude software from the patent paradigm suggests that subject matter is not a “failed gatekeeper”, but an important mechanism by which “soft” issues such as ethics and social considerations, can be reconciled within patent law. Finally, some remarks are addressed to a broader issue. As it is suggested that software ought not to be protected by patent law, the implications of that position on the protection of software is considered. It will be suggested that there is no need to warp patent law protect the industry. Alternative protection mechanisms are explored, and found to be better suited to crafting an Information Age response to an Information Age problem.

3 Other matters

The law stated herein is correct up to the end of June 2011. Where relevant, decisions made since that time have been noted. In particular, the decisions in *CyberSource Corporation v Retail Decisions Inc*³² and *Association for Molecular Pathology v USPTO*,³³ are briefly noted, but not comprehensively analysed. As to Australian law, despite it being nearly 5 years since the Federal Court has given judicial consideration to the scope of subject matter, it seems that Australian law will continue to track developments in the US, as it has in the past. For example, the *CyberSource* case was applied by the Australian Patent Office only days after being handed down.³⁴ *Bilski* itself was first applied in the APO decision in *Invention Pathways*³⁵ only a month after

³² Appeal No 2009-1358 (Fed Cir, 2011).

³³ Appeal No 2010-1406 (Fed Cir, 2011).

³⁴ See *Network Solutions, LLC* [2011] APO 65 (19 August 2011).

³⁵ *Invention Pathways Pty Ltd* [2010] APO 10 (21 July 2010).

Introduction

being handed down by the US Supreme Court. In Europe, the impasse between the UK courts and the EPO has not yet been resolved, although plans to introduce a unitary patent system would mean the EPO wins by default. It seems like the 2005 battle over the Software Patent Directive is about to resume in a different form.³⁶

Some parts of this thesis were originally drawn from the following published works:

1. Anton R Hughes, "A Comment on Software, Patents, Innovation and Openness" (2003) 14(1) *Journal of Law, Information and Science* 50;
2. Anton R Hughes, "Avoiding the Software Patent Problem: An Alternative Fix for TRIPS Junkies" (2007) 14(1) *eLaw: Murdoch University Electronic Journal of Law* 100;
3. Anton R Hughes, "Submission to the Advisory Council on Intellectual Property: Patentable Subject Matter Issues Paper" (19 September 2008)
<<http://www.acip.gov.au/reviewpatentable/IP22%20Anton%20Hughes.pdf>>; and
4. Anton R Hughes, "Case Brief: *Bilski v Kappos*" (2010) 20 *Journal of Law, Information and Science* 229.

³⁶ See Chapter 3, at 117. On the current battle, see for example Richard Stallman, "Beware: Europe's 'unitary patent' could mean unlimited software patents" *The Guardian*, 22 August 2011
<<http://www.guardian.co.uk/technology/2011/aug/22/european-unitary-patent-software-warning>> (11 September 2011). Further information is available at <<http://www.unitary-patent.eu/>> (11 September 2011).

The Nature of Software

1 Introduction

To understand why software is an issue for patent law, it is first necessary to understand a number of things about how modern software is organised, and how any particular software component is likely to be built. The nature of software will be returned to in Chapter 7, in much greater detail. However, an understanding of the software development process is necessary in order to properly understand the software patent case law in Chapter 3, and the problems which attend on these cases, set out in Chapter 4.

Through an exploration of the nature of software, four things will become clear. Firstly, the process of abstraction away from specific hardware limitations has been the key to the success of software. Secondly, the way in which software systems are constructed using layers, libraries, frameworks and design patterns makes it clear that one important goal of software development is to encourage re-use. Thirdly, the development of a particular piece of code tends to be very context-specific – not all software development activities are alike. Finally, software entities tend to be complex, typically involving large numbers of unique components. Abstraction, re-use, context and complexity all raise important considerations for the patentable subject matter issue which will be developed in later chapters.

This chapter will also introduce one of the major themes of the thesis – that software is isomorphic to mathematics. This isomorphism can be demonstrated both at the formal level, and at the structural level. Whilst there are undoubtedly some differences between the two, the *prima facie* identity established provides adequate grounds for exploring software's patentability by reference to the patentability of mathematics.

2 What is software?

Early in the history of computing, collection of physical components which make up a computer began to be called "hardware". Software was then coined to refer to the non-physical

Chapter 1: The Nature of Software

aspects of the computer, especially the programs, or sets of instructions “which cause a computer to perform a desired operation or series of operations.”¹ Such a definition is useful to a certain extent, but overlooks the complex relationship of hardware and software, particularly when a previously externally-stored program² is loaded into the memory of a computer. It also encourages a focus on the end product of the software development process – an executable manifestation of software – at the expense of other artefacts which exist at intermediary stages, determine the final product, and should probably fall within the definition of software.

So to truly understand the nature of software, it is necessary to see it in the broader context of computer science. First of all, it is necessary to understand how the development of software has evolved from modern times, as much of this evolution has determined the way in which software developers currently operate. With this in mind, it will then be possible to demonstrate the way in which software developers take an abstract idea and develop it into working software, through the paradigm known as top-down programming. It will also be possible to make some general observations about what makes software unique.

The Evolution of Modern Software

Machine Code

Machine code, or native code, is the medium in which all software was originally written, and in which software ultimately executes on a computer. It is also the only language which can be understood and executed by a computer, even now. Machine code instructions are a string of 1s and 0s (or bits) of a set length which tell the computer’s central processing unit (CPU) which instruction to perform, where to get the data, and where to put the result. An example of a 32-bit instruction is given below:³

1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0
Instruction (ADD)								Output (L4)				Input 1 (L2)				Input 2 (L3)				(unused)									

Table 1.1: Example machine code instruction

This instruction tells the CPU to add the contents of memory location 2 to memory location 3 and put the result in memory location 4. Writing whole programs in machine code would obviously be very tedious and time consuming. Further, errors are easy to make and hard to find. Finally, the set of instructions available tend to vary from computer to computer, meaning

¹ “Software” in *The Macquarie Dictionary*, (4th ed, The Macquarie Library Pty Ltd, 2003).

² Such as on punched cards, a floppy disk, or in more recent times, a CD or DVD.

³ Adapted from Arvind Asanovic, “Early Developments: From the Difference Engine to IBM 701” (Lecture Slides, MIT, 2005) <http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-823-computer-system-architecture-fall-2005/lecture-notes/101_earlydev.pdf> (11 September 2011).

there is little portability.⁴ So early on, programmers began to look for ways of moving towards more human-readable representations of computer instructions.

Assembly Language

The first milestone achieved by programmers in their move away from machine code was to write a program which could convert to and from a symbolic representation of the instruction. This program was called an assembler/disassembler. The assembly language version of the machine code instruction above looks like this:

```
ADD (L4, L2, L3)
```

“ADD” is obviously more memorable than 100000, but there is still a one-to-one correspondence between assembler instructions and machine code instructions. Seemingly simple operations require a number of machine-level instructions,⁵ and assembly programmers still have to work within the constraints of hardware-specific instruction sets.⁶ So they set to work on hiding the low-level details of the computer in another way.

High level languages

Most programming in contemporary times is done using high level languages. These languages are another step closer to natural language, and a typical instruction in a high level language may be made up of a series of machine-level instructions. High level languages also hide the machine architecture from the programmer. High level languages can be broadly categorised as one of two types, compilers or interpreters. A high-level representation of the example above would be:

```
a = b + c
```

In other words, when using high level languages, programmers no longer have to worry about memory locations, architecture-specific instructions or other low-level hardware details. However, in order to be executed by the computer, high-level language commands need to be translated into machine code. As such, high level languages are often classified according to how this translation is achieved, either by compiler or interpreter.

⁴ This was a particularly big issue early in the history of computing, where there was no standardisation of architecture. Although it may appear to be less of an issue now, where most computers are based on the Intel Pentium architecture, there are still many alternative architectures in use, especially for smaller electronic devices like PDAs and mobile phones.

⁵ The following example of assembler code for multiplying two numbers on an AVR chipset requires no less than 30 separate instructions. See Gerhard Schmidt, “Mult8.asm” on *Tutorial for learning assembly language for the AVR-Single-Chip-Processors* <http://www.avr-asm-tutorial.net/avr_en/calc/MULT8E.html> (24 July 2008).

⁶ Despite the difficulties, assembly language is still used today in applications where efficient performance in the software is valued over the manual labour involved in creating it, for example in device drivers, embedded systems and computer game routines. See Wikipedia, “Assembly language” <http://en.wikipedia.org/wiki/Assembly_language> (18 August 2008).

Chapter 1: The Nature of Software

Compilers

Compiled languages cannot be run directly on a computer in their original (source code) form. Before they can be executed, a translation program (compiler) translates the high-level language instructions to machine-specific instructions (the result of compilation being known as object code). Common compiled languages include C and Java.

Interpreters

A more modern development are interpreted languages, for which there is only source code. When these programs are run, a program called an interpreter translates high-level code to machine code as the code is run. Modern examples of interpreted languages include Python, Perl, PHP and Ruby. The downside of interpreted languages is that they generally run slower and require more memory. However, since no compilation is involved, the development and testing of software written in these programs is much more efficient.

But of course, the abstraction of the software development process does not end there. The efficiency of software development to this point involved abstracting software away from the low level details of the hardware on which it was to be run. A similar type of abstraction can be achieved by breaking software down into a series of layers, where each layer offers the layer above access to its services, but only via a set of higher-level functions. The typical layers involved in the operation of a modern computer are discussed below.

Firmware, kernels and operating systems

Firmware exists right at the borderline between hardware and software, and can be thought of as a computer program embedded in a hardware device. The most familiar role for firmware in modern computers is as the Basic Input Output System (or BIOS) which launches a computer's startup process by detecting the installed hardware elements, and starting to load the operating system. The reader might be able to just think of it as the first screen which appears when a computer is turned on.

Operating just above the level of the firmware is an operating system's kernel. The kernel handles the lowest level of interactions with the hardware of a computer. The most commonly known kernel is the Linux, the kernel of a GNU/Linux operating system. The job of the kernel is to "manage the computer's resources and allow other programs to run and use these resources"⁷ These resources include the CPU, memory (RAM) and various input/output devices such as displays, disk drives, mice and keyboards. Since there is usually more than one program running at any time on a computer, the kernel can be thought of as deciding which program gets access to what, when, and for how long. The kernel also takes care of communication between programs.

⁷ Wikipedia, "Kernel (computer science)" <[http://en.wikipedia.org/wiki/Kernel_\(computer_science\)](http://en.wikipedia.org/wiki/Kernel_(computer_science))> (23 July 2008).

Other parts of the operating system perform a wider range of higher level services, including:⁸

- program execution (starting and stopping applications)
- responding to events (e.g. “somebody hit the “A”key – send an A to the current program!”)
- memory management (e.g. “Firefox opened a new window – give it more memory!”)
- multitasking (e.g. surfing the web, running a mail merge, printing and drinking coffee at the same time)
- disk access and file systems
- device drivers (e.g. making sure Times New Roman looks the same on the printer at work as it does on the printer at home)
- networking
- security (e.g. who can log in, who can see this file)
- graphical user interface (windows, buttons, menus and so on)

Libraries, frameworks and patterns

Since the history of computer science is a lesson in the benefits of building on what has come before, software developers usually look to break software down into sets of components which can be re-used beyond the problem at hand. As such, the development of most, if not all, applications are built using a series of independent components, which are often written by independent developers, or as a collaborative effort between various parties.

A library (or module) is a “a substantial number of computer program modules designed to solve a wide range of problems in [a] given area”.⁹ For example, a library might contain a functions for calculating mathematical or scientific formulae,¹⁰ or functions for managing files.¹¹ Libraries allow programming effort to be concentrated on efforts that can be re-used. Libraries can be thought of as extending a language.

A framework is “the skeleton of an application that can be customized by an application developer”¹² and generally provide a set of reusable components such as libraries, as well as a design or design or pattern by which to combine them to form a whole application. Frameworks consist of both code, in the form of components, as well as a method of organising those

⁸ This list was adapted from Wikipedia, “Operating system” <http://en.wikipedia.org/wiki/Operating_system> (23 July 2008).

⁹ “Software Libraries, Numerical and Statistical” in Anthony Ralston, Edwin D Reilly and David Hemmendinger (eds), *Encyclopedia of Computer Science* (4th ed, Wiley, 2003).

¹⁰ For the C language, common mathematical functions are available in the “math” module, distributed with the compiler. A list of numerical and scientific libraries (or modules as they are called in Python) are available at <<http://wiki.python.org/moin/NumericAndScientific>> (29 July 2008). In the Perl language, CPAN.

¹¹ For example, “io.h” in C, “sys” in Python, or “csv” in Python for interpreting comma separated value files.

¹² Ralph E Johnson, “Frameworks = (components + patterns)” (1997) 40(10) *Communications of the ACM* 39 at 39.

Chapter 1: The Nature of Software

components into a larger whole. As an organising concept, they begin to stray beyond the realm of software as “code” into the realm of the purely abstract.

Patterns, or design patterns, are a “general reusable solution to a commonly occurring problem in software design”.¹³ The original concept of patterns was imported to the software field from architecture,¹⁴ and allows a design to be built from a catalogue of generalised solutions to recurring problems. A design pattern “describes the problem, the solution, when to apply the solution, and its consequences”.¹⁵ Since the mid 1990s,¹⁶ design patterns have had an increasingly influential role in the development of software.

The level of abstraction at which patterns exist is such that it becomes questionable whether they are software at all. As a formal manifestation of an process, they have a lot in common with the formal manifestation of an idea in a high level language. They are certainly a useful design tool, and also provide a vocabulary for describing the way software works. Intuitively speaking though, it is submitted that unless they are realised in an ultimately executable form,¹⁷ they should not be considered to be software.

Scripting languages and domain-specific languages

Applications themselves are also frequently developed in such a way that they can be extended by plug-in modules. Scripting languages are programming mini-languages which are primarily designed for controlling software applications. They allow users to write “programs within a program”, usually to automate repetitive tasks within the program, rather than having any broader role in software development. A common example would be Microsoft Word macros, written in a subset of the Visual Basic example. Javascript is another example of a scripting language, which is embedded in web browsers and is used to make traditionally static web pages more dynamic, or, dare I say it, more Web 2.0.

Similarly, domain-specific languages are designed to solve a set of needs specific to a particular problem domain. For example, Structured Query Language, or SQL, is the language used to handle interactions with most databases.¹⁸ Cascading Style Sheets are another example, and are widely used to centrally manage the formatting of web sites (font size, background colours and so on).¹⁹

¹³ Wikipedia, “Design pattern (computer science)” <[http://en.wikipedia.org/wiki/Design_pattern_\(computer_science\)](http://en.wikipedia.org/wiki/Design_pattern_(computer_science))> (29 July 2008).

¹⁴ See Christopher Alexander et al, *A Pattern Language: Towns, Buildings, Construction* (Oxford University Press, 1977).

¹⁵ Erich Gamma et al, *Design Patterns: Elements of Reusable Object-Oriented Software* (Addison-Wesley Professional, 1994).

¹⁶ The take-off in design patterns is generally attributed to the publication of the so-called “Gang of Four” book, Erich Gamma et al, *Design Patterns: Elements of Reusable Object-Oriented Software* (Addison-Wesley Professional, 1994).

¹⁷ By this it is meant that they should be manifested in either machine code, assembly language or a compilable or interpreted high-level language.

¹⁸ For an introduction see John Worsley and Joshua Drake “Chapter 3: Understanding SQL” in *Practical PostgreSQL* (O’Reilly, 2001) <<http://www.faqs.org/docs/ppbook/c1164.htm>> (3 November 2008).

¹⁹ For an introduction see Dave Raggett “Introduction to CSS” <<http://www.w3.org/MarkUp/Guide/Style>> (4 November 2008).

The modern software development process

With a general understanding of how software development takes place at differing levels of abstraction, and in different contexts, it is now important to understand how programmers write it. The process “is one of gradual refinement and elaboration”,²⁰ starting with a highly abstract concept, namely the pure idea and moves, through a series of steps, towards specific code – often seen as the the medium in which modern programmers work. Ogilvie identifies 6 individual levels of abstraction within the software development cycle,²¹ and whilst many of them may take place in parallel, they are helpful in explaining how software developers ‘bridge the gap’ from concept to code.²²

To give a more concrete sense of how these levels of abstraction influence each other, the discussion will discuss a fictional program to manage a dental prosthetics business, called OverByte.²³

Main purpose

What is meant by the “main purpose” of a computer program is a high-level description of the problem which the software is intended to solve.

At its simplest, the main purpose of OverByte is that noted above, namely to manage a dental prosthetics business, although what would typically be involved would be a more detailed statement of the problems which the package is supposed to solve, such as tracking, managing and reporting on the ordinary transactions engaged in by a dental prosthetics business on a day-to-day basis.

System architecture

As the programmer learns more and more about the problem to be solved, they will begin to break the problem up into a series of smaller problems, or modules. “Each module performs a significant portion of the program’s main purpose and is eventually implemented as a distinct section of the source code.”²⁴ The system architecture describes how the program is dissected into these modules, and the interactions and/or relationships between them. The interaction between modules can be described by the way they ‘call’ each other (control flow), the way they push data to each other (data flow) or by the way they are ‘nested’ inside each other (to make an analogy, a ‘family’ module might have 2 ‘parent’ modules and 2 ‘children’ modules).

For OverByte, the modules involved might be grouped into something like order entry, invoicing, accounts receivable, end of day procedures and end of month procedures.

²⁰ Peter G Spivack “Comment: Does Form Follow Function? The Idea/Expression Dichotomy in Copyright Protection of Computer Software” [1988] 35 *UCLA Law Review* 723 at 729.

²¹ John WL Ogilvie, “Defining Computer Program Parts Under Learned Hand’s Abstractions Test in Software Copyright Infringement Cases” (1992) 91 *Michigan Law Review* 526.

²² It should be noted that the process involved describes the modern paradigm of object-oriented programming.

²³ The example is borrowed (but modified) from the US case of *Whelan Associates v Jaslow Dental Lab Inc* 797 F.2d 1222 (1986).

²⁴ John WL Ogilvie, “Defining Computer Program Parts Under Learned Hand’s Abstractions Test in Software Copyright Infringement Cases” (1992) 91 *Michigan Law Review* 526 at 534.

Chapter 1: The Nature of Software

Abstract data types (ADTs)

Abstract data types, also known as objects, define both the sorts of data stored (data type), as well as the operations which can be performed on it. “[T]he motivation behind every ADT is to associate a given data type with the operations that are useful in manipulating that type”.²⁵

An example for the OverByte program might be an invoice data type. An invoice would store data such as:

- invoice number
- invoice date
- due date
- items
- subtotal
- GST
- Total

Operations which could be done on an invoice would include such things as:

- print
- update invoice values
- delete the invoice
- calculate GST
- process invoice payment

Algorithms and Data Structures

Algorithms and data structures are more computer-specific representations of operations and data types than just discussed. It is important to note that what distinguishes algorithms and data structures from ADTs is that there may be a number of specific algorithm and data structure combinations that correctly implement the requirements of the ADT. “Although data structures and algorithms are more specific than ADTs and may even depend on certain programming language features for use, they are independent of any specific programming language or piece of literal source code.”²⁶ **Algorithms**, also known as procedures, are a ‘unit’ of computation which perform a specific task. Whilst an “operation merely identifies a desired result,

²⁵ John WL Ogilvie, “Defining Computer Program Parts Under Learned Hand’s Abstractions Test in Software Copyright Infringement Cases” (1992) 91 *Michigan Law Review* 526 at 536.

²⁶ John WL Ogilvie, “Defining Computer Program Parts Under Learned Hand’s Abstractions Test in Software Copyright Infringement Cases” (1992) 91 *Michigan Law Review* 526 at 540

What is software?

an algorithm specifies every step necessary to accomplish that result”.²⁷ Typically, procedures take data as an input, and act on it to create a new set of data as an output. For example, the following represents a procedure for sorting a set of numbers from lowest to highest:

Procedures are usually only considered at this somewhat abstract level, so as not to be constrained by the specific limitations of either computer hardware and a particular programming language, although their operation may be described in more detail than here, either in natural language, a flowchart, or in some sort of pseudo-code.

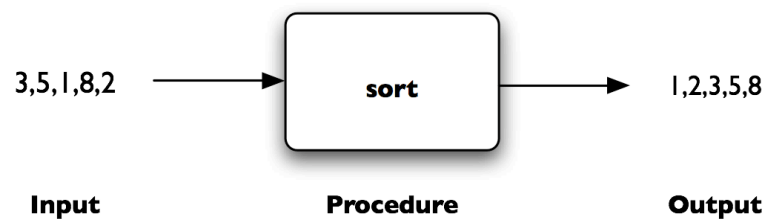


Figure 1.1: Sort procedure.

A simple example from the OverByte program’s invoice data type would be an algorithm to calculating the GST on an invoice. Such an algorithm might work as follows:²⁸

1. Add up the total cost of items on the invoice.
2. Divide the total by 10, rounding up to 2 decimal places
3. Save this value into the GST field.

Data structures describe how a computer is to represent/store the data in a data type. A data structure “consists of one or more variables of the basic data types, which are organized in some specified combination of arrays, records, and pointers”.²⁹ Data structures are defined at an abstract level which does not require any consideration of the way they will be represented in the memory of a computer. Considered together, procedures and data spaces can be thought of as a preliminary sketch of a program and are usually written in natural language, or some formal notation.

Many of the data structures in the OverByte invoice data type are simple data structures. For example, the subtotal, GST and total would be stored as floating point (i.e. decimal) numbers. The “items” data is a little more complicated, in that it is an “array of records”. Arrays and records differ in that with an array, each sub-component is the same data type. With a record, each sub-component may be of a different type. Considering the invoice items data structure

²⁷ John WL Ogilvie, “Defining Computer Program Parts Under Learned Hand’s Abstractions Test in Software Copyright Infringement Cases” (1992) 91 *Michigan Law Review* 526 at 536.

²⁸ The savvy reader may wonder why the GST isn’t just calculated from the subtotal field. The author admits that the algorithm has been chosen for its illustrative nature, rather than its computational (or logical) efficiency. In other words, calculating it from the subtotal seems too trivial to deserve the label ‘algorithm’.

²⁹ John WL Ogilvie, “Defining Computer Program Parts Under Learned Hand’s Abstractions Test in Software Copyright Infringement Cases” (1992) 91 *Michigan Law Review* 526 at 539.

Chapter 1: The Nature of Software

as a table, each row is a record, because individual columns are of different types, as shown below:

Quantity	Description	Unit price	Item cost
integer/whole number	string/text	floating point/decimal	floating point/decimal

Table 1.2: “Items” data structure

Although only one row is shown, multiple rows of data could exist on an actual invoice. The data structure for each column is also shown. Each row will have the same columns, so will be an array of records.

Source code

With a preliminary design in place, the next step is for the procedures and data spaces to be written out in the programmer’s programming language of choice. The programmer is limited in the expression of the code at this stage by both the syntax and grammar of the code. Depending on context and choice of language, the hardware on which the program is to run may also influence the way the source code is written. Because it is in many ways the final stage of the development process, the software reflects all the previously discussed levels of abstraction. As Ogilvie notes:

A program’s entire range of abstraction is embedded in its code in roughly the same way a novel’s characters and plot are embedded in its text. Therefore, care must be taken to avoid confusing code as the embodiment of parts at every level of abstraction with code as a level of abstraction in its own right.³⁰

A brief extract of how the Invoice components of OverByte thus far discussed might look if written in the Python programming language might look is included in Algorithm 1.1 on the facing page. The code should illustrate that source code is designed as much for humans as for computers. Even those unfamiliar with programming should be able to follow what is going on in this code, particularly aided by the comments embedded throughout, preceded by the # character. The significance of this point cannot be understated, and will be returned to at various points in this thesis..

Object code

This is the final, executable form of software. The transition from source code to object code is usually an automated process, although it may involve some degree of manual configuration

³⁰ John WL Ogilvie, “Defining Computer Program Parts Under Learned Hand’s Abstractions Test in Software Copyright Infringement Cases” (1992) 91 *Michigan Law Review* 526 at 541.

```

from datetime import datetime

class Invoice(BaseContent):

# set up data structures
def __init__(self):
    self.invoiceNumber = self.getInvoiceNumber()
    self.invoiceDate = datetime.date() # today's date
    self.dueDate = datetime.date() + 14 # 2 weeks from today
    self.items = [] #array in which to put item records
    self.subtotal = 0.0
    self.gst = 0.0
    self.total = 0.0

    # code to add an item to the invoice
    def addItem(self, quantity, description, unit_price):
        item = {} # set up row (record)

        item['quantity'] = quantity
        item['description'] = description
        item['unit_price'] = unit_price
        item['item_cost'] = quantity * unit_price

        # add record to items array
        self.items.append(item)

        #update invoice totals
        self.subtotal = self.subtotal + item['item_cost']
        self.gst = self.calculateGST(self)
        self.total = self.subtotal + self.gst

# code to calculate GST
def calculateGST(self):
    subtotal = 0

    #add up total cost of items
    for item in self.items:
        subtotal = subtotal + item['item_cost']

    gst = subtotal * 0.1
    self.gst = round(gst, 2) # round to 2 decimal places

```

Algorithm 1.1 The OverByte Invoice class

before it begins by the programmer. As noted above in relation to interpreted languages, the distinction between source code and object code can be very subtle indeed. In fact, the code set out above would be converted to machine code by the interpreter as it is run.

Chapter 1: The Nature of Software

Putting it all together

What the discussion thus far should make clear is that the development of software products is dependent on a series of factors. These will now be discussed in turn.

Abstractness Firstly, the history of computer science sees more and more abstraction away from the hardware on which the computer runs, getting ever closer to the abstract realm of ideas. Assembly language is an abstraction of machine code, and high level languages are a further abstraction of assembly language. Each of the layers in the development stack above can also be seen as an abstraction away from the specifics of the layer below.

Secondly, this abstraction means that software is “pure thought-stuff, infinitely malleable.”³¹ Not only can it be changed however, it is also “embedded in a cultural matrix of applications, users, laws, and machine vehicles [which] all change continually, and their changes inexorably force change upon the software product.”³²

Another aspect of abstractness which can lead to characterisation difficulties, is that different levels of abstraction may sit on different conceptual layers. This is most obvious when considering high level languages. On one conceptual level, the software can be considered to be a description of the functionality that the program will have when it is executed. At the same time, the software *causes* the executing computer to exhibit that behaviour. These conceptual layers are reflected to some extent in compiled languages, where the source code is descriptive and the object code is functional. But such neat divisions can be misleading. Even machine code is descriptive to a programmer who understands it. Further, with interpreted languages, the source code *is* the object code, and is thus both descriptive and functional at the same time.

Further, the abstract nature of software means that it is invisible, or intangible. Unlike other abstractions, such as a floor plan of a building however, software abstractions are “not inherently embedded in space. Hence, [they have] no ready geometric representation in the way that land has maps, silicon chips have diagrams, and computers have connectivity schematics.”³³ Various diagrams are used to attempt to represent software,³⁴ however these fail to capture the whole of what software is.

Finally, intangible creations such as software are not subject to the myriad physical uncertainties³⁵ which normally have to be overcome as a part of the experimentation process, since they are “largely determined, in advance, by the specification, the flow chart, the rules of the

³¹ Frederick P Brooks Jr, “No Silver Bullet: Essence and Accidents of Software Engineering” (1987) 20(4) *Computer* 10 at 12.

³² Frederick P Brooks Jr, “No Silver Bullet: Essence and Accidents of Software Engineering” (1987) 20(4) *Computer* 10 at 12.

³³ Frederick P Brooks Jr, “No Silver Bullet: Essence and Accidents of Software Engineering” (1987) 20(4) *Computer* 10 at 12.

³⁴ For example, flow of control, the flow of data, patterns of dependency, time sequence, and name-space relationships. See Frederick P Brooks Jr, “No Silver Bullet: Essence and Accidents of Software Engineering” (1987) 20(4) *Computer* 10 at 12.

³⁵ See Jay Dratler Jr, “Does Lord Darcy Yet Live? The Case Against Software and Business Method Patents” (2003) 43 *Santa Clara Law Review* 823 at 854, and also Chapter 6 at 265.

programming language, the programming conventions, and the dictates of logic and mathematics.”³⁶

As a result, context is also very important when discussing software. The context in which a software component sits is key to determining how much it has to do with the hardware on which it runs, and the dependence which a product has on the work which has gone before. For example, writing firmware components in assembler is a task which is largely constrained by the hardware configuration that it is to be written for, and will probably rely less on reusable components than will writing a plug-in macro for a word processor.

Re-use The development of software from machine code on also indicates that progress in software depends on re-use of the work of the past – assemblers were written in machine code, compilers were written in assembly language and so on. It is through abstraction from the specific to the general that much of this re-use can be achieved.

Re-use also influences the design of software by encouraging conformity. As a corollary, the expectation of conformity from users also encourages re-use. This expectation is often formalised into the notion of a standard.³⁷ Standards exist for such things as file formats,³⁸ user interfaces³⁹ and network communication protocols.⁴⁰

Complexity Another factor which may not be entirely apparent from the discussion above (although the layers in Illustration 5 below may give a clue) is its complexity. As Brooks notes, “[s]oftware entities are more complex for their size than perhaps any other human construct because no two parts are alike. If they are, we make the two similar parts into a subroutine”.⁴¹ Taking this absence of repetition into account, the number of lines of code in a program can provide a rough estimate of the magnitude of complexity of software products. For example, version 2.6.24 of the Linux kernel, which alone comprises only a small part of a GNU/Linux

³⁶ Jay Dratler Jr, “Does Lord Darcy Yet Live? The Case Against Software and Business Method Patents”(2003) 43 *Santa Clara Law Review* 823 at 855.

³⁷ Standards can be developed privately or unilaterally, and may become a standard through custom or convention (a de facto standard, such as the Microsoft Word document format) by mandate (such as when a government body demands compliance with a certain standard) or formal consensus (such as through the formal vote of a standards organisations like the International Standards Organisation). They often take the form of “a formal document that establishes uniform engineering or technical criteria, methods, processes and practices”: Wikipedia, “Standard” *Wikipedia.org* <<http://en.wikipedia.org/wiki/Standard>> (3 October 2008).

³⁸ Aside from the de facto standard of the Word document noted above, the alternative OpenDocument Format was passed by the International Standards Organisation in 2006. See “Open Document Format for Office Applications (OpenDocument) v1.0” ISO/IEC 26300:2006 <http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=43485> (3 October 2008).

³⁹ See for example Apple Inc, “Introduction to Apple Human Interface Guidelines” [2008] *Apple Developer Connection* <http://developer.apple.com/documentation/UserExperience/Conceptual/AppleHIGuidelines/XHIGIntro/chapter_1_section_1.html> (2 October 2008).

⁴⁰ The protocols used for the internet, commonly referred to as TCP/IP were made the standard for all military computer networking by the US Department of Defense in March 1982. See Wikipedia, “TCP/IP” <<http://en.wikipedia.org/wiki/TCP/IP>> (8 October 2008). The official specification is documented in the following document: Internet Engineering Task Force, “Request for Comments 1122: Requirements for Internet Hosts” <<http://tools.ietf.org/html/rfc1122>> (8 October 2008).

⁴¹ Frederick P Brooks Jr, “No Silver Bullet: Essence and Accidents of Software Engineering” (1987) 20(4) *Computer* 10 at 12.

Chapter 1: The Nature of Software

operating system, contains over 8.5 million lines of code.⁴² Red Hat 7.1, a GNU/Linux distribution released in 2001 contained around 30 million lines of code.⁴³ Windows XP, released in October 2001, contains around 40 million lines of code.⁴⁴

However, one must be careful in noting this complexity, because there is a simplicity about software which is further unlike physical objects.⁴⁵ Software components are stable and predictable and lacking in uncertainty. Their behaviour can be predicted beforehand with complete accuracy, unlike for example, the momentum and location of an electron in orbit around an atom.⁴⁶

3 An example

As time has gone on, and layer after layer of abstraction has been introduced, the distance the connection between many programmers' sphere of interest has about as much to do with the hardware of a particular computer as an author's over-cafeinated state has to do with the coffee machine with which his third cup of coffee was made.

For example, for a number of years this author has been developing web applications based on an open source content management system called Plone. One such project was for an online education site which needed the ability to create and administer multiple choice tests through the web. The online test program itself ran on top of a large stack of abstractions, as shown in the diagram below:⁴⁷

The online test product relied on components belonging to the Plone system. Plone itself is written in a framework called Zope. Zope, Plone and the test product are all written in the interpreted language Python, which must be configured and compiled for the particular operating system on which Zope is to be run. The layers from the operating system down have been discussed already. If even a few layers were not present, authoring such an online test product would have been so costly as to be prohibitive. Thus the economics of software development depends on re-using these abstractions for its continued viability, since writing online test components from the hardware up in machine code would be far too laborious.

Combining this notion of context with the process of software development described above, it is then possible to better illustrate how abstractness and context are important when discussing the appropriate level of protection for software:

⁴² Greg Kroah-Hartman, Jonathan Corbet and Amanda McPherson "Linux Kernel Development (April 2008)" [2008] *The Linux Foundation* <<http://www.linuxfoundation.org/publications/linuxkerneldevelopment.php>> (28 August 2008).

⁴³ David Wheeler, "Counting Source Lines of Code (SLOC)" <<http://www.dwheeler.com/sloc/>> (28 August 2008).

⁴⁴ Larry O'Brien, "How Many Lines of Code in Windows?" on *Knowing.NET* <<http://www.knowing.net/PermaLink,guid,c4bdc793-bbcf-4fff-8167-3eb1f4f4ef99.asp>> (28 August 2008).

⁴⁵ This point is aptly illustrated by Jay Dratler Jr, "Does Lord Darcy Yet Live? The Case Against Software and Business-Method Patents" (2003) 43 *Santa Clara Law Review* 823 at 864-9.

⁴⁶ So says Heisenberg's uncertainty principle, which lies at the heart of quantum physics. See Jan Hilgevoord and Jos Uffink "The Uncertainty Principle" *Stanford Encyclopedia of Philosophy* <<http://plato.stanford.edu/entries/qt-uncertainty/>> (21 October 2008).

⁴⁷ This diagram is adapted from the computer architecture abstraction layer diagram on Wikipedia, "Kernel (computer science)" <[http://en.wikipedia.org/wiki/Kernel_\(computer_science\)](http://en.wikipedia.org/wiki/Kernel_(computer_science))> (12 November 2008).

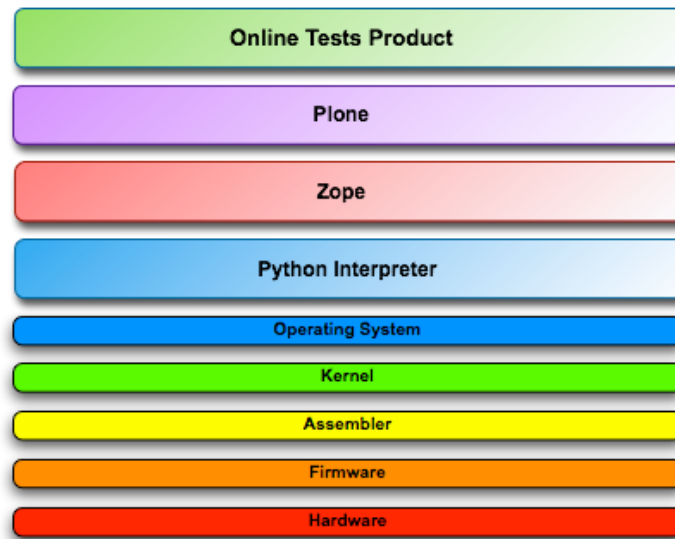


Figure 1.2: Online test development stack

Of course the diagram is idealised in certain respects. Often, the software development process takes place on various levels at once, so it can be difficult to isolate these layers from each other. Often the various layers evolve together. Further, various components of a particular program may involve development at different layers of the stack. In the diagram for example, it is possible that some components may bypass the Plone layer, and interact directly with Zope or Python.

But despite these simplifications, the diagram does illustrate a point. To judge the level of abstractness of a particular software component, it is then necessary to know where the component sits in the scheme of layers of abstraction. Further, it is necessary to know at what level of the software development process protection is being afforded.

The individual layers are on close inspection made up of further layers. For example, the operating system can be broken down into the layers shown in Table 1.3.⁴⁸

Command interpreter
Long-term scheduler
Resource manager
Short-term scheduler
File manager
I/O system
Memory manager
Kernel

Table 1.3: Operating system layers

⁴⁸ The table was taken from Larry L Wear et al, *Computers: An Introduction to Hardware and Software Design* (McGraw-Hill, 1991) at 104.

Chapter 1: The Nature of Software

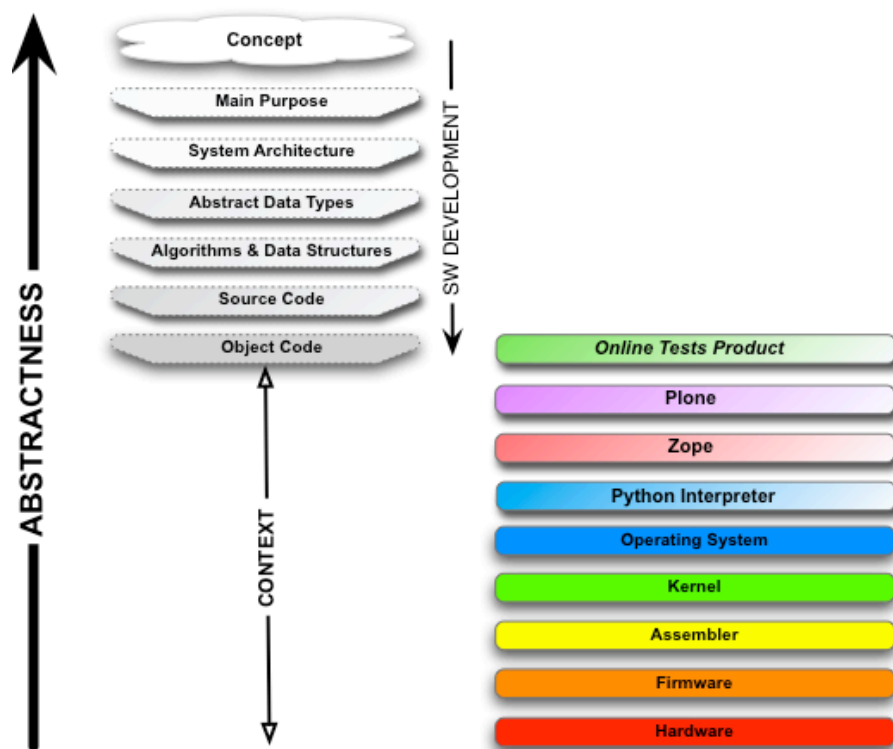


Figure 1.3: Abstractness and context in software development

These are the dimensions of software seen from the perspective of computer science and software development. But this is not the whole story. Hiding within the story of the history of software is an important relationship. The lowest level of computing, machine code, is a number cruncher. Both the computer instructions, and the data on which it acts are represented as a series of numbers – 1s and 0s. The available instructions are themselves a set of ways of transforming these numbers. Numbers and their transformations are clearly the domain of mathematics, which suggests at least a similarity, and at most an identity. So, are mathematics and software wholly identical? If not, what are the dimensions of their relationship? This question will now be addressed.

4 How is software related to mathematics?

Most attention given by patent law scholars to understanding the relationship between mathematics and software has focused on the patentability of algorithms. Whilst an algorithm can be defined broadly to include any conceivable process, courts have called on the notion of a *mathematical* algorithm, which has been defined as “a procedure for solving a given type of mathematical problem”.⁴⁹ However, since discussion above makes it clear that algorithms are only really one component of software, such an understanding may seem an insufficient basis

⁴⁹ *Gottschalk v Benson* 409 US 63 (1972); *IBM v Commissioner of Patents* (1991) 33 FCR 218 at 220.

How is software related to mathematics?

by which to understand the relationship. Courts considering the copyright of software have not been so restrained. There are two reasons why the algorithm has taken centre stage, one historical, one practical. Historically, software was written in a way different to that described earlier in this chapter – programs were written as one long sequence of code, or one algorithm. This remained the dominant approach until the late 1980s, when the top down methodology took hold. However, all of the US Supreme Court cases on the patentability of software took place before this point. But this historical justification is limited in its usefulness, since there have been a substantial number of cases in other courts since this time, none of which have considered this change. Further, the same courts when considering copyright have demonstrated a clear understanding of that change.⁵⁰

Practically then, the reason algorithms are so important is because it is primarily in algorithms that the function, rather than the structure of software is to be found. Since functionality is what the patent regime tries to protect, it is to be expected that courts have kept algorithms at the centre of their focus.⁵¹ The organisation of code, although important forms part of the “mere scheme or plan”⁵² for carrying out the implementation of the software by programmers, and is thus beyond the operation of patent law.⁵³ Thus it is both sufficient and useful to consider the relationship of software and mathematics through the lens of the algorithm.

Algorithms, as a formalised description of a process, give a clue as to the relationship of software and mathematics. In the early part of the 20th century, the mathematician David Hilbert sought to prove that the truth of mathematical principles could be formally demonstrated.⁵⁴ A formal proof is in effect a formalised version of the mathematical algorithm as described above. A formalised notion of mathematics seeks to prove the truth of a mathematical theorem as follows:

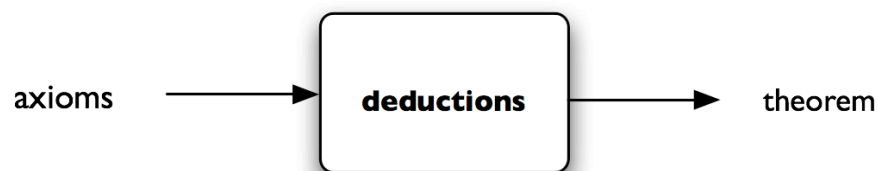


Figure 1.4: Deducing theorems from axioms

The formalist deductive process takes as its starting point a set of foundational truths, or axioms.⁵⁵ Each step of the deductive process involves the application of any one of a set of rules,

⁵⁰ See for example the progression from the approach of the court in *Whelan Associates v Jaslow Dental Lab Inc* 797 F.2d 1222 (1986) to that in *Computer Associates International, Inc. v Altai, Inc.* 982 F.2d 693 (1992).

⁵¹ It should be noted however that data structures have sought to be protected as well. See for example *In re Alappat* 33 F.3d 1526 (1994); *Welcome Real-Time v Catuity* (2001) 113 FCR 110.

⁵² *Cooper's Application* (1902) 19 RPC 53.

⁵³ Alternatively, it may be considered part of the non-patentable idea rather than a substantive part of the invention. Finally, the organisation of the code may be too far removed from the “useful product” required by the court in *Grant v Commissioner of Patents* [2006] FCAFC 120.

⁵⁴ Hilbert's formalist programme is discussed in greater detail in Chapter 4.

⁵⁵ Exactly what is to be considered axiomatic can itself be a deep philosophical issue. This will be explored in more

Chapter 1: The Nature of Software

whose truth can also be considered axiomatic. The idea is that if you agree with the truth of the starting point, and the truth of each step along the way, it follows that the truth of the destination must also be accepted. Thus any theorem that can be deduced from the axioms using the rules of the system, then the theorem must also be true. Although the formalist programme was ultimately unsuccessful, the notion of the reduction of reasoning to a series of mechanical steps lead to a number of formal models of computation, which despite their diversity have been shown to be equivalent “in the sense that each analysis offered has been proved to pick out the same class of functions”⁵⁶ The most famous of these is the Turing machine. The formal equivalence of mathematics, the Turing machine and modern computers is considered below.

Formal equivalence: the Turing machine

The Turing machine was conceived by Alan Turing in 1936 as a way of formal statement of computability, in other words, a definition of what could be computed. At the heart of this definition was a device called the Turing machine. A Turing machine consists of the following components:

1. a control unit, which can assume any one of a finite number of possible states;
2. [an infinitely long] tape, marked off into discrete squares, each of which can store a single symbol taken from a finite set of possible symbols; and
3. a read-write head, which moves along the tape and transmits information to and from the control unit”⁵⁷

On each iteration of a cycle, the Turing machine’s next action will be determined by a table of “transition rules”.⁵⁸ For example, in the image above, the symbol ‘A’ and the state ‘q3’ determine the next step taken by the machine. These actions might include:

- writing or erasing the current symbol in the current position of the tape;
- moving the tape left or right, or leaving the tape where it is; or
- changing the internal control state.⁵⁹

detail in Chapter 4. Russell and Whitehead used formalism in what might be considered to be reverse order, attempting to deduce the whole of mathematics from the rules and axioms of logic in order to prove that logic was the proper foundation of mathematics.

⁵⁶ Jack Copeland, “The Church-Turing Thesis” on *AlanTuring.net* (2000)

<http://www.alanturing.net/turing_archive/pages/Reference%20Articles/The%20Turing-Church%20Thesis.html> (accessed 22 May 2007). See also Todd Rowland, “Church-Turing Thesis” *Wolfram MathWorld* (2002) <<http://mathworld.wolfram.com/Church-TuringThesis.html>> (accessed 23 April 2007) at 3. For a comprehensive survey, see Stephen C Kleene *Introduction to Metamathematics* (North-Holland, 1980) at 12-13.

⁵⁷ “Turing Machine” in Anthony Ralston, Edwin D Reilly and David Hemmendinger (eds), *Encyclopedia of Computer Science* (4th ed, Wiley, 2003).

⁵⁸ David Barker-Plummer “Turing Machines” *Stanford Encyclopedia of Philosophy* <<http://plato.stanford.edu/entries/turing-machine/>> (19 July 2008).

⁵⁹ Alan M Turing, “On Computable Numbers, with an Application to the Entscheidungsproblem” (1937) 47 *Proceedings of the London Mathematical Society* 230.

How is software related to mathematics?

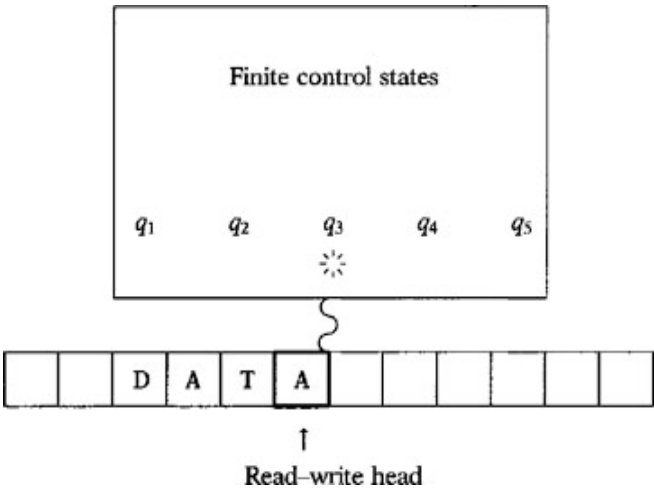


Figure 1.5: The Turing machine

When the machine reaches a state for which there is no transition rule (or more than one), it is taken to have completed its computations, and the machine halts. An example of a transition table is given below:⁶⁰

Configuration		Behaviour	
Control state	Tape symbol	Tape operations	Final control state
b	blank	Print "0", Move right	c
c	blank	Move right	e
e	blank	Print "1", Move right	z
z	blank	Move right	b

Table 1.4: A Turing machine transition table

This example works as follows. This program endlessly loops through each of the control states b,c,e and z in order, and can be represented diagrammatically as follows:

The effect on the tape is to print alternating ones and zeros with a blank space in between, as shown below:

0		1		0		1		0		1	...
---	--	---	--	---	--	---	--	---	--	---	-----

Table 1.5: Turing machine output tape

Recalling the make-up of a typical machine-language instruction in Table 1.1, the visual similarity with the above illustration gives a sense of how the Turing machine relates to modern computers. Despite the apparent simplicity of this machine, it is capable of much more complex calculations than the above example suggests. In fact, Turing proposed that “[Turing machines] can do anything that could be described as ‘rule of thumb’ or ‘purely mechanical’”.⁶¹ In other

⁶⁰ Adapted from Alan M Turing, “On Computable Numbers, with an Application to the Entscheidungsproblem” (1937) 47 *Proceedings of the London Mathematical Society* 230 at 233.
⁶¹ Alan M Turing, “Intelligent Machinery” in Jack Copeland (ed), *The essential Turing: seminal writings in computing, logic, philosophy* (Oxford University Press, 2004) at 414.

Chapter 1: The Nature of Software

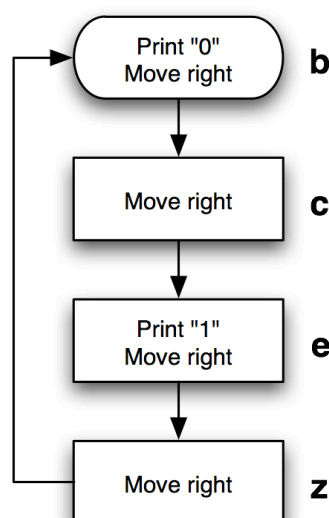


Figure 1.6: Turing machine control state flow diagram

words, “[w]henver there is an effective method for obtaining the values of a mathematical function, the function can be computed by a Turing machine.”⁶² This statement can not however be proved in a precise fashion because what is computable and what is not is something which is largely intuitive. As such it is more of a “working hypothesis”,⁶³ but nonetheless one which continues to be applicable. For this reason, the group of functions which are computable are also known as functions that are “intuitively calculable”(or computable).

It is however possible to note a common feature of all known calculable functions with more precision. Alonzo Church was able to show that they have the quality of being general recursive.⁶⁴ Church was also able to demonstrate that any such algorithms could be written in a mathematical notation called the lambda calculus (λ -calculus). The λ -calculus and Turing machine “turned out to be equivalent, in the sense that each picks out the same set of mathematical functions”.⁶⁵ As such, the notion that all known effectively calculable mathematical algorithms can be calculated by a Turing machine is known as the Church-Turing thesis.

⁶² B Jack Copeland, “The Church-Turing Thesis” *Stanford Encyclopedia of Philosophy* <<http://plato.stanford.edu/entries/church-turing/>> (20 July 2008).

⁶³ Emil L Post, “Finite Combinatory Processes - Formulation 1” (1936) 1 *Journal of Symbolic Logic* 103, cited in B Jack Copeland, “The Church-Turing Thesis” *Stanford Encyclopedia of Philosophy* <<http://plato.stanford.edu/entries/church-turing/>> (20 July 2008).

⁶⁴ A simple example of a partially recursive function is the Fibonacci number series (1,2,3,5,8,...), where the next number in the series is the sum of the previous two numbers in the series. Although the series is defined by reference to itself (this is what makes it recursive) we know that working out. say, the 15th element in the series can be broken down into a fixed sequence of steps. A general recursive function is also defined by reference to itself, but in such a way that it creates an infinite loop. The classic example of this is Epimenides’ liar paradox, which can be put in the form “This phrase is false”. To assess the truth value of the statement. It is interesting to note that Gödel’s Incompleteness Theorem relies on a variation of the Liar paradox. The Liar paradox, Gödel’s theorem and other “strange loops” are expertly dealt with in Douglas R Hofstadter *Gödel, Escher, Bach: An Eternal Golden Braid* (Basic Books, 1980).

⁶⁵ B Jack Copeland, “The Church-Turing Thesis” *Stanford Encyclopedia of Philosophy* <<http://plato.stanford.edu/entries/church-turing/>> (20 July 2008).

How is software related to mathematics?

Although the focus of the Church-Turing thesis is on determining the boundaries of computability, the formal equivalence of mathematics and software follows from the equivalence of λ -calculus and Turing machine programs. λ -calculus is clearly mathematics, and can be used to describe all known mathematical algorithms. Similarly, Turing machine transition tables are a form of software. Thus it follows that all computable functions can be solved either by mathematical methods, or by a Turing machine program.

But before a true equivalence can be reliably claimed, a further link must be made between real-world computers, and the abstract definition of a Turing machine. If real computers and Turing machines are equivalent, then it follows that software and mathematics must be identical, because anything that can be written in software, can be described in the λ -calculus.

Are computers and the Turing machine equivalent?

The Strong Church-Turing thesis holds that a Turing machine can do whatever a computer can do. A certain level of support for this proposition is to be found in similarities between the λ -calculus and programming languages. For example, the λ -calculus has been referred to as “the smallest universal programming language.”⁶⁶ The λ -calculus has also influenced the design of a number of functional programming languages such as Lisp, ISWIM and ML.⁶⁷

One obvious difficulty in accepting the equivalence of Turing machines and computers is the Turing machine’s unlimited storage. The importance of unlimited storage in the model is that it allows a broader notion of effective computability which “ensures that no computable function will fail to be Turing-computable solely because there is insufficient time or memory to complete the computation.”⁶⁸ For practical purposes, this difference can be worked around by assuming that the storage space of a real computer could be infinitely extended if required.⁶⁹ It would also be sufficient to show that anything calculable by a real world computer was at the very least a subset of the functions effectively calculable by a Turing machine since it would still lead to a conclusion that the overlap between mathematics and real world computers was a complete one.

The major difference between the Turing machine and modern computers is the way in which data is accessed. The Turing machine is a sequential-access machine, since it processes the memory (tape) one square at a time. Modern computer memory is not sequential. Any position

⁶⁶ Wikipedia, “Lambda calculus” <http://en.wikipedia.org/wiki/Lambda_calculus> (5 November 2008). For a thorough account of the use of the λ -calculus as a programming language itself, see John R Harrison “Chapter 3 - Lambda calculus as a programming language” *Introduction to Functional Programming* (Lecture Notes, Cambridge University, 1997) <<http://www.cl.cam.ac.uk/teaching/Lectures/funprog-jrh-1996/>> (5 November 2008).

⁶⁷ See John R Harrison “Chapter 3 - Lambda calculus as a programming language” *Introduction to Functional Programming* (Lecture Notes, Cambridge University, 1997) <<http://www.cl.cam.ac.uk/teaching/Lectures/funprog-jrh-1996/>> (5 November 2008) at 25-26.

⁶⁸ David Barker-Plummer “Turing Machines” *Stanford Encyclopedia of Philosophy* <<http://plato.stanford.edu/entries/turing-machine/>> (19 July 2008).

⁶⁹ Given the continued exponential growth in storage space evident throughout the whole history of computing, this does not seem an overly unrealistic assumption to make. For example, when the first hard drive was shipped by IBM in 1956, it was 5Mb in size. By 1980, IBM had introduced a 1,000Mb hard drive. By 2007, Seagate had built a 1,000,000 Mb hard drive. See Jon Wuebben, “The History of the Floppy and Hard Disk Drive” (20 March 2007) <<http://www.patantconsult.com/articlesvault/Article/The-History-of-the-Floppy-and-Hard-Disk-Drive/11501>> (11 September 2011) ; Wikipedia, “Moore’s Law”, <http://en.wikipedia.org/wiki/Moore%27s_law> (18 September 2010).

Chapter 1: The Nature of Software

in memory can be accessed at any time. A computer based on such a design is called a random access machine (RAM). Despite these design differences however, it has been shown that both designs are of equivalent power. Since a Turing machine can simulate a RAM, and a RAM can simulate a Turing machine,⁷⁰ it follows that anything which can be designed for one machine can be run on the other (via the simulator, if not directly). Since these machine-level simulations of a modern computer are possible on a modern computer, our discussion of the development of assembler and higher level languages above makes it clear that software written in modern computer languages can be made to run on a Turing machine, and vice versa, so all modern software is equivalent to Turing machine programs.

Criticisms of the Church-Turing thesis

The Church-Turing thesis has its critics who suggest that it may well owe more the “mathematics worldview”,⁷¹ prevalent at the inception of computer science as a discipline, rather than to any intellectual rigour. In particular, the definition of a Turing machine cannot account for interaction between the program and its environment, suggesting it “may no longer be fully appropriate to capture all features of present-day computing”⁷² such as artificial intelligence, graphics and the Internet. The difficulty with the Turing model, based as it is in mathematics, is in accounting for input after the program has started. As Tseytin puts it:

An algorithm in the mathematical sense is completely self-contained and as soon as the data have [sic] been specified it needs no further information. In contrast to this, a realistic procedure (and, to some extent, a modern computer program) can draw information from the environment in a way that *need not be specified in advance*.⁷³

Despite this, the continued applicability of the Turing model is illustrated by the fact that the Turing model can be extended rather than replaced in order to overcome such limitations.⁷⁴ Whilst these extended models might not be equivalent to the original, there is currently no evidence to suggest they are more powerful. So the Turing machine still remains central in the theory of computer science. In any event, there is nothing about these extensions that necessarily breaks the equivalence between mathematics and software, since there is no reason why similar extensions could not be made to mathematics to encompass interactivity. In any event, an algorithm which collects all data beforehand is functionally equivalent to an algorithm which collects data interactively so any difference does not alter the formal equivalence of mathematics and software.

⁷⁰ Marvin L Minsky, *Computation: finite and infinite machines* (Prentice Hall, 1967), Chapter 11.

⁷¹ Dina Q Goldin and Peter Wegner “The Church-Turing Thesis: Breaking the Myth” in *Proceedings of CiE 2005* 152 at 154-155.

⁷² Jan van Leeuwen and Jiri Wiedermann, “The Turing Machine Paradigm in Contemporary Computing” in B Enquist and W Schmidt (eds), *Mathematics Unlimited - 2001 and Beyond* (Springer Verlag, 2000).

⁷³ Gregory S Tseytin “From Logicism to Proceduralism (An Autobiographical Account)” in *Algorithms in Modern Mathematics and Computer Science* (Springer Verlag: Berlin 1979) 390 at 395.

⁷⁴ See for example the Persistent Turing Machine put forward in Dina Q Goldin and Peter Wegner “The Church-Turing Thesis: Breaking the Myth” in *Proceedings of CiE 2005* 152. Van Leeuwen offers an Interactive Turing Machine for the same purpose – see Jan van Leeuwen and Jiri Wiedermann, “Beyond the Turing Limit: Evolving Interactive Systems,” in *SOFSEM 2001: Theory and Practice of Informatics* (2001) <http://dx.doi.org/10.1007/3-540-45627-9_8> (accessed 3 November 2008).

How is software related to mathematics?

Others have criticised the relevance of the Turing machine on the basis of industry practice. Dryja notes that the Turing machine fit neatly with computer software when it was primarily written by computer scientists, in a largely sequential form, almost a one-algorithm-per-program approach. Dryja claims that software development is now primarily the domain of engineers since it is centred around the top-down object-oriented programming approach described above. There is some truth in this contention, in that programs are not written as one long series of instructions any more. But objects themselves merely “perform computations and save local state”⁷⁵ are basically Turing machines. This in turn means that an object-oriented program is a collection of inter-related Turing machines. On this basis it is hard to accept that the Turing machine, and hence mathematics is no longer relevant to an understanding of what software is. A collection of mathematical objects is no less mathematical in nature than its constituent parts.

What the above criticisms do suggest is that the Turing machine model is not a complete account of what can be computed on a modern day computer. Despite these criticisms however, the fact remains that Turing machines remain fundamental in importance, since “they are widely used and have been widely adopted as the standard model in computability theory.”⁷⁶ Thus despite these criticisms, it is clear that the formal equivalence of software and mathematics, via the Turing machine, can nonetheless be reliably claimed.

Still, in the unlikely event any particular criticism has left the reader in any doubt as to the equivalence of mathematics, there is other evidence to support the claim of equivalence at a structural level. This structural equivalence is considered below.

Structural similarities between mathematics and software

Beyond the Turing machine, there are structural similarities which suggest an equivalence between software and mathematics. As opposed to the formal equivalence noted above, the focus in determining whether there is a correspondence between structures found in software development and those used in mathematical activities”.⁷⁷ If the activities engaged in by software developers is isomorphic, then the two are structurally equivalent. The starting equivalence is known as the Curry-Howard isomorphism,⁷⁸ which notes a correspondence between programming and mathematical logic. This isomorphism is a natural result of the formal equivalence of formal systems and the Turing machine noted above. Kondoh summarises it as follows:⁷⁹

⁷⁵ Mark Stefik and Daniel G Bobrow, “Object-Oriented Programming: Themes and Variations,” (1986) 6(4) *AI Magazine* 40 at 41.

⁷⁶ VJ Rayward-Smith *A First Course in Computability* (Blackwell Scientific, 1986) at vii.

⁷⁷ Hidetaka Kondoh, “What Is ‘Mathematicalness’ in Software Engineering?” in *Proceedings of Fundamental Approaches to Software Engineering 2000*, 163 at 170.

⁷⁸ The full isomorphism is a combination of three seminal works by Curry and Howard. See Haskell Curry, “Functionality in Combinatory Logic” (1934) 20 *Proceedings of the National Academy of Sciences* 584; Haskell Curry and Robert Feys, *Combinatory Logic, Volume 1* (North-Holland, 1958) at paragraph 9E; William A Howard, “The formulae-as-types notion of construction”, in Jonathan P Seldin and J Roger Hindley (eds), *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, (Academic Press, 1980) 479.

⁷⁹ Hidetaka Kondoh, What Is ‘Mathematicalness’ in Software Engineering? at 170.

Chapter 1: The Nature of Software

Programming	Logic
Specification	Theory
Program	Proof

Table 1.6: The Curry-Howard isomorphism

In a similar vein, the relationship between software and formal systems can be seen by comparing the definitions of programming languages and formal systems. First consider the following definition of a programming language:

A set of *symbols*, understood by both sender and receiver, is combined according to a set of rules, its grammar or syntax. The semantics of the language defines how each grammatically correct sentence is to be interpreted.⁸⁰

The similarity is clear when compared to the following definition of a formal system:

Each formal system has a formal language composed of primitive *symbols* acted on by certain *rules* of formation (statements concerning the symbols, functions, and sentences allowable in the system) and developed by inference from a set of axioms. The system thus consists of any number of formulas built up through finite combinations of the primitive symbols—combinations that are formed from the axioms in accordance with the stated rules.⁸¹

The problem with the isomorphism at this level is that it assumes that programs can be “systematically derive[d] ... from their specifications”,⁸² an approach which meets the same criticisms Dryja levelled at the Turing machine style of programming, namely, that this just isn’t how programmers work any more. However, Kondoh is able to use the Curry-Howard isomorphism to demonstrate a wider equivalence between the modern art of software engineering and mathematics.⁸³

This similarity of structure between mathematics has led some computer scientists to apply mathematical proof techniques to computer programs in order to allow the verification of their results, and confirm that they meet their specifications.⁸⁴ This sort of translation takes place in the other direction as well, with computers having become an indispensable tool by which

⁸⁰ Alice E Fischer and Frances S Grodzinsky, *The Anatomy of Programming Languages 1* (1993). Cited in Gregory Stobbs, *Software Patents* (2nd ed, Aspen, 2000).

⁸¹ Encyclopaedia Britannica, “formal system” *Encyclopaedia Britannica Online* <<http://www.britannica.com/EBchecked/topic/213751/formal-system>> (28 June 2011).

⁸² Hidetaka Kondoh, “What Is ‘Mathematicalness’ in Software Engineering?” in *Proceedings of Fundamental Approaches to Software Engineering 2000*, 163 at 170.

⁸³ Hidetaka Kondoh, “What Is ‘Mathematicalness’ in Software Engineering?” in *Proceedings of Fundamental Approaches to Software Engineering 2000*, 163 at 171. See also Peter Suber, “Formal Systems and Machines: An Isomorphism” <<http://www.earlham.edu/~peters/courses/logsys/machines.htm>> (accessed 27 June 2011).

⁸⁴ See for example John McCarthy, “Towards a mathematical science of computation” in CM Popplewell (ed), *Information Processing 1962* (Holland Publishing Company, 1962) 21; <<http://www-formal.stanford.edu/jmc/towards.html>> (21 July 2008); C A R Hoare “An axiomatic basis for computer programming”(1969) 12(10) *Communications of the ACM* 576.

Implications of the isomorphism for patent law

Programming	Mathematics
Basic Control Structure (Repetitive Loop, Conditional, etc)	Elementary Proof Step (Mathematical Induction, Case Analysis, etc)
Idiom (Useful Combination of Control Structures)	Proof Technique (Conventional Technique for Fragmental Proof)
Abstract Data Type (Collection of Operations on Common Data)	Theory on a Mathematical Notation (Collection of Lemmata on a Mathematical Notion)
Design Pattern (Specific Combination of (Possible) Classes and Specific Use of their Interdependencies)	Proof Tactics (Specific Combination of Subgoals (Lemmata) and Specific Use of their Interdependencies)
Architectural Pattern (Specific Combination of Specification of Components)	Theory Strategy (Collection of Basic Definitions and the Main Theory)
Software System	Mathematical Theory (Structure formed by Definitions, Theorems and Proofs)

Table 1.7: Kondoh's wider programming-mathematics isomorphism

mathematicians test theories through modelling⁸⁵ and the automation of formal proofs.⁸⁶ That this can be done only strengthens the intuitive sense that there is an intimate connection between the two disciplines.

5 Implications of the isomorphism for patent law

The underlying premise of patent law, whatever the justification,⁸⁷ is its claimed net-positive impact on innovation. Broadly speaking, innovation is “the implementation of a new or significantly improved product (good or service), process, new marketing method or a new organisational method in business practices, workplace organisation or external relations”.⁸⁸ As such, it is asserted that deeper understanding of the impact of patent law on innovation can be arrived at by considering the nature of the activities undertaken by the relevant innovators, that is, mathematicians and programmers. Since the two domains share a formal and structural equivalence, it follows that the effect of external factors such as the patent regime on innovative activities in these fields is likely to be, *prima facie*, equivalent.

It is therefore on this basis that this thesis draws upon the treatment by patent law of mathematics, to justify a similar treatment for software. It should be stated however that it is not asserted that mathematics and software are in all respects identical, only that they are closely

⁸⁵ For an introduction to the area see Wikipedia, “Computer simulation” <http://en.wikipedia.org/wiki/Computer_simulation> (4 November 2008).

⁸⁶ For example, the four colour theorem, which states that any map can be coloured using no more than four colours without any two adjacent regions being the same colour, was the first major theorem to be proved by a computer. For an introduction to the field, see Wikipedia, “Interactive theorem proving” <http://en.wikipedia.org/wiki/Interactive_theorem_proving> (4 November 2008). For an ambitious project to build an automatically-verified wiki of mathematical knowledge, see Cameron Freer, “What is vdash?” <<http://www.vdash.org/intro/>> (11 September 2011).

⁸⁷ See Introduction, at 5. Whilst the general justification discussed in this thesis is the social contract theory of patent law, this is not the only justification. Nonetheless, the notion that patents create or encourage innovation or at least the commercialisation of research, is a dominant claim of patent law.

⁸⁸ OECD, Oslo Manual Guidelines for collecting and interpreting innovation data, (3rd edition, 2005). Cited Department of Innovation, Industry, Science and Research, “Introduction” *Australian Innovation System Report 2011*

Chapter 1: The Nature of Software

related. The similarities and differences between the two will become apparent later in this thesis, as the nature of the creative process in mathematics and software are considered in much greater detail in Chapters 6 and 7.

6 Conclusion

Two key themes have arisen from the discussion in this chapter. Firstly, the history of software and the nature of software development have made it plain that three key considerations dictate the dimensions by which software component must be studied. Successive iterations in the history of software have been delivered by layer **abstractions** upon further abstractions, to the point where many programmers are no longer concerned in any meaningful fashion with the specific physical characteristics of the hardware their components might run on. Because the many layers of abstraction involve different levels of interaction with the physical hardware of the computer, consideration of a software component must consider its **context** in order to properly account for the physical constraints affecting the component's design, and the extent to which the component builds on the work of others. From the constant building new layers on previous layers developed a culture of **re-use**, which still pervades software development to this day, and without which the economics of software development would be unsustainable. Further, the **complexity** of software means that one application may be made up of a large number of interrelated but unique components, which for a large product may number into the millions. It will be shown in the next chapter that many of the problems which software has thrown up in various legal contexts can ultimately be traced to these considerations.

Secondly, one of the main arguments of this thesis, the equivalence of mathematics and software has been demonstrated. The isomorphic nature of software and mathematics has been demonstrated at both a formal and structural level. Both the nature of software itself, and the way in which programmers work is inherently mathematical in nature. In other words, either software developers are in some sense "doing mathematics" when they are writing software or alternatively, they are undertaking activities so similar in nature to mathematics that they give rise to the same sorts of policy considerations which apply to the activities of mathematicians. The patentability of software will be considered in later chapters via the lens of mathematics, and it will be demonstrated that the non-patentability of software follows from the non-patentability of mathematics.

The History of Patentable Subject Matter

*The law embodies the story of a nation's development through many centuries and it cannot be dealt with as if it contained only the axioms and corollaries of a book of mathematics. In order to know what it is, we must know what it has been, and what it tends to become.*¹

*I do not find it at all easy to reconcile all the cases which have been cited to me, and all the observations made in the judgments in these cases.*²

1 Introduction

In the last chapter, a number of the fundamental characteristics of software were discussed. In particular the importance which considerations such as context, abstraction and reuse play in understanding the nature of any software component was noted. This chapter will lay out the parameters of the inherent patentability inquiry, through a historical account of the development of jurisprudence on the issue. Having discussed both the nature of software, and the purpose of the inherent patentability requirement, it will then be possible to properly understand why the patenting of software is problematic, both for the software industry, and for the patent system more generally.

In this Chapter it will be shown that a number of analytical tools have been used to help define the borderline between what is patentable and what is not. A common factor has been the contrast between abstract, intangible subject matter which is not patentable, and those involving a corporeal, tangible manifestation which are. The leading case in Australia on patentable subject matter is *National Research Development Corporation v Commissioner of Patents*³ ("NRDC"). The case has been described as a watershed⁴ and is sometimes characterised as a break with

¹ Oliver W Holmes Jr, *The Common Law and Other Writings* (Little, Brown and Co, 1881) at 1.

² *Re GEC's Application* (1941) 60 RPC 1 at 4 per Morton J.

³ (1959) 102 CLR 252.

⁴ *Joos v Commissioner of Patents* (1972) 126 CLR 611 at 616 per Barwick CJ. See also *CCOM Pty Ltd v Jiejing Pty Ltd* (1994) 51 FCR 260 at 287.

Chapter 2: The History of Patentable Subject Matter

the past.⁵ However, it will be argued that rather than representing a new phase in the development of patent law, *NRDC* was intended as a continuation of existing precedent, and that any claim to a “formula” arising from the case is a misapplication of its text.

The nature of the inherent patentability test is therefore part historical inquiry. A question thus arises as to how far back such a history should go. Given that computer software is a relatively recent phenomenon,⁶ it might be tempting to say “not far”. However, the relevant period of assessment depends not on the timeline of computing history, but on the time over which principles relevant to the inherent patentability requirement have been developed.⁷ This chapter traces the development of such principles from the earliest known patent customs in Europe through to the most recent developments in three jurisdictions – the United Kingdom, the United States of America, and Australia. The reasons for considering jurisdictions beyond Australia are as follows. Firstly, the Australian patent system is directly dependent on the jurisprudence developed in the UK before 1977. Secondly, despite claims that patent law is an instrument of national economic policy,⁸ Australia is not a heavily litigated jurisdiction. As a result, the development of the law in Australia has been influenced to a great degree by what is happening beyond these shores. American jurisprudence has been particularly influential in the development of Australian patent law at least over the last 50 years, since, as will be seen, developments over that time have broadly tracked those in the US. Thirdly, the adoption by the UK of the European Patent Convention since 1977 means that subsequent case law in that country provides a counterpoint to the US approach, and another (at least theoretical) source of influence on Australian patent law. In particular however, the developments in Europe in recent years in relation to a proposed software patent directive illustrate that the software patent debate is still a live one, rather than being a glitch in the system which has been subsequently corrected.

What is a patent?

According to IP Australia, a patent is “a right granted for any device, substance, method or process which is new, inventive and useful”.⁹ Unlike copyright, which protects expressions in a particular form, patents provide protection against functional equivalents and are therefore a stronger form of protection than copyright. The strength of the protection afforded is balanced

⁵ See for instance William van Caenegem, “The Technicality Requirement, Patent Scope and Patentable Subject Matter in Australia” (2002) 13 *Australian Intellectual Property Journal* 41; *CCOM Pty Ltd v Jieying Pty Ltd* (1994) 51 FCR 260 at 287: “[NRDC] changed the direction of case law not only in Australia but also in the United Kingdom.”

⁶ The term “software” itself was only coined in 1958: see Ivars Peterson “Software’s Origin” *Ivar Peterson’s MathTrek Archive*, 31 July 2000 <http://www.maa.org/mathland/mathtrek_7_31_00.html> (10 March 2010); David Leonhardt, “John Tukey, 85, Statistician; Coined the Word ‘Software’”, *New York Times* (New York), 28 July 2000, at A19. The origins of software are discussed in the context of the history of computing in Chapter 3.

⁷ The High Court in *NRDC* make it clear that what is patentable subject matter depends on answering the question “Is this a proper subject of letters patent according to the principles which have been developed for the application of section 6 of the Statute of Monopolies?”: (1959) 102 CLR 252 at 269. See Section 6 below.

⁸ “It is generally acknowledged that an effective intellectual property system plays a central role in building a strong national innovation system which is a key component of Australia’s economic development.”: Advisory Council on Intellectual Property, *Patenting of Business Systems: Issues Paper* (2002) at 1. Whether it is actually effective for such a purpose “is a matter of sustained, complicated and acrimonious controversy”: Patricia L Loughlan, “Patents: Breaking Into the Loop” (1998) 20 *Sydney Law Review* 553 at 555.

⁹ IP Australia, “Glossary” <<http://www.ipaustralia.gov.au/resources/glossary.shtml>> (25 February 2010).

by the shorter length of protection which patents provide. In Australia, two types of patents are available. There is the standard patent, which grants a monopoly to a successful applicant for up to 20 years. An innovation patent is a shorter, less expensive variation, with a maximum lifespan of 8 years.

In order to be granted a patent, an applicant must disclose the workings of their invention, and demonstrate that the claimed invention meets the requirements of patentability. These requirements are set out in s18(1) *Patents Act 1990* (Cth) as follows:

Subject to subsection (2), an invention is a patentable invention for the purposes of a standard patent if the invention, so far as claimed in any claim:

- (a) is a **manner of manufacture** within the meaning of section 6 of the Statute of Monopolies; and
- (b) when compared with the prior art base as it existed before the priority date of that claim:
 - (i) is **novel**; and
 - (ii) involves an **inventive step**; and
- (c) is **useful**; and
- (d) was **not secretly used** in the patent area before the priority date of that claim by, or on behalf of, or with the authority of, the patentee or nominated person or the patentee's or nominated person's predecessor in title to the invention.

Subsection (2) specifically excludes "human beings and the processes for their generation" from the realm of patentable subject matter.

What is a manner of manufacture?

The manner of manufacture test, also referred to as inherent patentability, or the subject matter inquiry, differs from the other patentability requirements in that it is applied not in relation to the characteristics of any claimed invention, but it operates to exclude whole categories of subject matter from the realm of patent law. As such, it tends to stray from the comfortable realm of factual questions such as "was the essence of this invention known by others at the priority date?" into the realm of policy-laden questions such as "is this invention a proper subject for the grant of a patent?".¹⁰

The term manner of manufacture appears (to the uninitiated at least) to give a clue as to the scope of "proper subjects" by reference to a literal interpretation of the term manufacture. Indeed the term patent may evoke images of industrial revolution era machines, or technological gadgets. However, as will be seen, determining the scope of patent law is a complex difficult question, and one which the judiciary have grappled with for well over 400 years.

¹⁰ *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 269. The question is discussed in the context of this case in Section 6.

Chapter 2: The History of Patentable Subject Matter

2 Early history

The grant of monopolies to encourage trade and establish new industries has a long history. The earliest recorded grant of such a patent is believed to have been made in Venice early in the 15th century, with many other European states having granted similar rights in the 15th and 16th centuries.¹¹ It has also been suggested that the English monarchy had exercised the Royal Prerogative to confer monopolies on trade guilds, corporations and individuals, by charters and patents, in order to establish new industries around the middle of the 16th century.¹²

But the history of the grant of these monopolies has had, in England at least, a rather chequered history. Despite the use of such grants as a means of encouraging industry, such monopolies came to be awarded by the Crown on an increasing basis as personal favours to courtiers, to “secure for them a source of income at the expense of the community.”¹³ Seen as an abuse of power, and interference with trade, the award of such “odious monopolies” became part of the struggle for power between the Crown and the House of Commons. This struggle eventually led to the passage of the *Statute of Monopolies*¹⁴ which declared “all monopolies and commissions, grants, licences, charters and letters patent for the sole buying, making working or using of anything within this Realm ... utterly void and of none effect.”¹⁵ Section 6 of the statute contained an exception:

Provided also (and be it declared and enacted) that any declaration before mentioned shall not extend to any letters patent and grants of privilege for the term of fourteen years or under, hereafter to be made, of the sole working or making of any manner of new manufactures within this realm, to the true and first inventor and inventors of such manufactures which others at the time of making such letters patent and grants shall not use, so as also they be not contrary to the law or mischievous to the state, by raising prices of commodities at home, or hurt of trade, or generally inconvenient.

Not only does the *Statute of Monopolies* serve to illustrate that patent law has always been an exception to a general prohibition on monopolies, it also continues to form the basis of the Australian test for patentable subject matter. The current test, set out above, requires that a patentable invention be “a manner of manufacture within the meaning of section 6 of the Statute of Monopolies”.¹⁶ That this test can remain relevant nearly 400 years later has little to

¹¹ See Edward C Walterscheid, “The Early Evolution of the United States Patent Law: Antecedents (Part 1)” (1994) *Journal of the Patent and Trademark Office Society* 697 at 708 (footnote 37).

¹² Edward C Walterscheid, “The Early Evolution of the United States Patent Law: Antecedents (Part 2)” (1994) 76 *Journal of the Patent and Trademark Office Society* 849 at 850; Peter Meinhardt, *Inventions, Patents and Monopoly* (2nd ed. 1950) at 43; Klaus Boehm, *The British Patent System, Volume 1: Administration* (1967). Cited in Sam Ricketson and Miranda Richardson, *Intellectual Property: Cases, Materials and Commentary* (3rd ed. 2005) at 14-26.

¹³ Peter Meinhardt, *Inventions, Patents and Monopoly* (2nd ed. 1950) at 43.

¹⁴ *Statute of Monopolies* 1623, 21 Jac 1, c 3 (UK) (“Statute of Monopolies”)

¹⁵ *Statute of Monopolies*, s.1.

¹⁶ *Patents Act* 1990 (Cth), s.18(1)(a).

do with the verbal formulation of the section, and more to do with the judicial development of the concept of manner of manufacture since that time.¹⁷

The passage of the *Statute of Monopolies* is sometimes treated as the beginning of the history of patent law. However, as noted above, the grant of patents predated the passing of the Statute, and as such the Statute is really a codification of the patent custom in force at that time.¹⁸ Sherman and Bently note that this elevation of the Statute did not take place until the nineteenth century, at a time when “Whig interpretations of the constitution [in which Parliament, not the Crown, was the source of ‘official’ power] ... gained widespread currency.”¹⁹ This statute-centric view has no-doubt been reinforced by the practice of making statutory definitions of patentable subject matter depend on the Statute since 1883.²⁰

One illustration of the Statute as a mere Code is found in Coke’s explanation of the phrase “contrary to law” in his *Institutes of Law*.²¹ Coke linked this formulation to the precedent of *Bircot’s case*,²² decided in 1572, where it was “resolved, that if the new manufacture can be substantially invested according to law, yet no old manufacture in use before can be prohibited.”²³ Despite the meaning such words appear to have, the phrase represents perhaps the earliest recognisable exception to patentable subject matter, a prohibition on patents for improvements to an existing manufacture. According to Coke, the subject of the patent had to be “substantially and newly invented: if the substance was *in esse* before, and a new addition thereunto, though that addition make the former more profitable, yet it is not a new manufacture in law”.

From this prohibition on derivative subject matter, it can also be inferred that patents were only being awarded for “entire trades and devices”,²⁴ giving a clue as to the nature of invention at that time. It is also clear that the notion of invention also extended to the importation of goods into the realm.²⁵ In fact, this was the main type of “invention” for which patents were then awarded.²⁶

¹⁷ Ricketson and Richardson note that the term “has come to assume a life and meaning of its own, one that flows far beyond the boundaries of expected and conventional meaning.”: Sam Ricketson and Miranda Richardson, *Intellectual Property: Cases, Materials and Commentary* (3rd ed, LexisNexis 2005) at 14-26.

¹⁸ Hulme noted in 1900 that “the effect of the statute is essentially to confirm the practice of the Crown during the period A.D. 1561 to 1603”: E. Wyndham Hulme, “The History of the Patent System Under the Prerogative and at Common Law – A Sequel” (1900) 16 *Law Quarterly Review* 44 at 44. See also Justine Pila, “The Common Law Invention in its Original Form” [2001] *Intellectual Property Quarterly* 209; Edward C Walterscheid, “The Early Evolution of the United States Patent Law: Antecedents (Part 2)” (1994) 76 *Journal of the Patent and Trademark Office Society* 849 at 849-850.

¹⁹ Brad Sherman and Lionel Bently, *The Making of Modern Intellectual Property Law* (1999) at 209 (footnote 14). In particular, Sherman and Bently note that in the Report of the 1829 Select Committee on Patents, the Statute of Monopolies was considered of minimal importance, but “by 1891 Robert Frost was able to say without hesitation in his text on patent law that the Statute of Monopolies was the statutory foundation of modern patent laws”: *Ibid.*

²⁰ Discussed below.

²¹ Sir Edmund Coke, *The third part of the Institutes of the Laws of England* (1644).

²² Sir Edmund Coke, *The third part of the Institutes of the Laws of England* (1644) at 184.

²³ Sir Edmund Coke, *The third part of the Institutes of the Laws of England* (1644) at 184.

²⁴ E. Wyndham Hulme, “The History of the Patent System Under the Prerogative and at Common Law – A Sequel” 16 *Law Quarterly Review* 44; Justine Pila, “The Common Law Invention in its Original Form” [2001] *Intellectual Property Quarterly* 209.

²⁵ Christine MacLeod, *Inventing the Industrial Revolution: The English patent system, 1660-1800* (1988) at 18; Edward C. Walterscheid “The Early Evolution of the United States Patent Law: Antecedents (Part 2)” (1994) 76 *Journal of the Patent and Trademark Office Society* 849 at 877.

²⁶ According to Walterscheid, the establishment of local industry based on continental technologies was a key factor in the award of monopolies during Elizabethan times. See Edward C. Walterscheid “The Early Evolution

Chapter 2: The History of Patentable Subject Matter

Beyond this codification, passage of the Statute was an important development in its own right:

Its importance lay in the crystallization, in giving letters patent for invention statutory recognition and hence legal status; perhaps it even allowed them a new lease of life when the Crown might have lost interest in issuing them. But in thus codifying, the Act made barely perceptible alterations in the administration of the patent system, and in practice left jurisdiction over it with the Privy Council, where it remained until 1753.²⁷

3 Post-1623 developments

In the years subsequent to the passage of the Statute, little further development of patent law took place. This can be explained for two reasons. Firstly, the Privy Council retained jurisdiction over all matters relating to patents until 1753, and thus any challenge to a patent grant involved a direct challenge to the authority of the Crown in exercising the Royal Prerogative.²⁸ Secondly, the attitude of the Crown was predisposed towards patent grants, since it was considered to be “at the hazard of the petitioner both as to his expenses and trouble whether the said Invention have the success he expects or not”.²⁹

By the middle of the 18th Century, the nature of technology began to change, and patents began to be sought and awarded for more abstract subject matter, namely, “technical improvements on existing products and processes”.³⁰ The common law courts, having taken over the settlement of patent disputes from the Privy Council began to adapt the law to suit the times. The prohibition on patents for improvements which had influenced patent law since 1572 was overturned in *Morris v Bramson*³¹ in 1776, recognising administrative practice over the previous century.³² In particular Lord Mansfield noted that to hold that patents for improvements were invalid “would go to repeal almost every patent that was ever granted”.³³

Boulton v Bull

A typical example of this new breed of invention based on improvement was the issue of a patent to James Watt in 1769 for a major improvement to the steam engine which had the effect

of the United States Patent Law: Antecedents (Part 2)” (1994) 76 *Journal of the Patent and Trademark Office Society* 849 at 855-857.

²⁷ Christine MacLeod, *Inventing the Industrial Revolution: The English patent system, 1660-1800* (1988) at 19.

Walterscheid points out that “giving legal status” might be an overstatement, but agrees that it laid a “firm foundation for the development of patent law”: Edward C. Walterscheid “The Early Evolution of the United States Patent Law: Antecedents (Part 2)” (1994) 76 *Journal of the Patent and Trademark Office Society* 849, at 880.

²⁸ Edward C Walterscheid “The Early Evolution of the United States Patent Law: Antecedents (Part 3 continued)” (1995) 77 *Journal of the Patent and Trademark Office Society* 847 at 848.

²⁹ D Seaborne Davies, “Early History of the Patent Specification” (1934) 50 *LQR* 86 at 87. Also noted in Edward C Walterscheid “The Early Evolution of the United States Patent Law: Antecedents (Part 3 continued)” (1995) 77 *Journal of the Patent and Trademark Office Society* 847 at 848.

³⁰ Justine Pila, “Inherent patentability in Anglo-Australian law: a history” (2003) 14 *Australian Intellectual Property Journal* 109 at 114-115.

³¹ *Morris v Bramson* (1776) 1 Carp Pat Cas 30; 1 Abbot’s PC 21 (KB).

³² Edward C Walterscheid “The Early Evolution of the United States Patent Law: Antecedents (Part 3 continued)” (1995) 77 *Journal of the Patent and Trademark Office Society* 847, at 852.

³³ *Morris v Bramson* (1776) 1 Carp Pat Cas 30 at 34.

of diminishing the consumption of fuel. This patent became the subject of dispute in what may be the most important early patentable subject matter case of all, *Boulton v Bull*.³⁴ The Court was constituted by 4 judges, all of whom determined the matter from different perspectives. The case is an early illustration of how difficult it is to reconcile competing accounts and considerations of patent law in a consistent fashion.

Principles

One point of agreement between all members of the Court was that bare principles are non-patentable. According to Buller J, the non-patentability of principles flows from “the very statement of what a principle is”,³⁵ namely “the first ground and rule for arts and sciences, or in other words the elements and rudiments of them”.³⁶ Eyre CJ was similarly disposed to hold that “[u]ndoubtedly there can be no patent for a mere principle”³⁷ or “abstract notion”³⁸ since “the carrying into execution of any new invention [must involve] certain means proper for the operation”,³⁹ which in turn required the invention to be “embodied and connected with corporeal substances . . . and to produce effects in any art, trade, mystery or manual occupation”.⁴⁰ For Heath J, principles fell outside the two classes of patentable manufacture, namely “machinery [and] substances (such as medicines) formed by chemical and other processes”.⁴¹

Rooke J was more accommodating of the possibility that principles might be patentable. His Honour held that such principles as claimed in this case were widely known and hence did not form part of the inventive output of the claimant.⁴² This interpretation seems to hold open the possibility of patenting a “newly discovered natural principle as to steam, [or] any new mechanical principle”⁴³ although, it could equally be argued that his Honour’s definition of principles as “first principles”⁴⁴ precludes any requisite notion of novelty which might ever make them patentable.⁴⁵

Processes

In *Boulton v Bull*, the patentability of processes was also squarely raised as an issue. Watt had chosen to claim his invention as “a new invented method of using an old engine in a more beneficial manner than was before known”,⁴⁶ rather than as a machine incorporating such an improvement. Watt, who was quite knowledgeable as to patent law, understood the

³⁴ *Boulton & Watt v Bull* (1795) 126 ER 651, hereinafter referred to as *Boulton v Bull*.

³⁵ *Boulton v Bull* (1795) 126 ER 651 at 662.

³⁶ *Boulton v Bull* (1795) 126 ER 651 at 662.

³⁷ *Boulton v Bull* (1795) 126 ER 651 at 667.

³⁸ *Boulton v Bull* (1795) 126 ER 651 at 667.

³⁹ *Boulton v Bull* (1795) 126 ER 651 at 666.

⁴⁰ *Boulton v Bull* (1795) 126 ER 651 at 667.

⁴¹ *Boulton v Bull* (1795) 126 ER 651 at 660.

⁴² *Boulton v Bull* (1795) 126 ER 651 at 659.

⁴³ *Boulton v Bull* (1795) 126 ER 651 at 659.

⁴⁴ *Boulton v Bull* (1795) 126 ER 651 at 659.

⁴⁵ Justine Pila takes this view. See Justine Pila, “Inherent patentability in Anglo-Australian law: a history” (2003) 14 *Australian Intellectual Property Journal* 109 at 117. Note that such a view is also consistent with the description of Buller J of principles as the “first ground . . . for arts and sciences” discussed above.

⁴⁶ *Boulton v Bull* (1795) 126 ER 651 at 651.

Chapter 2: The History of Patentable Subject Matter

importance of seeking a patent for the method, rather than just the improved machine due to the broader scope of protection which such a patent would have.⁴⁷ Such an interpretation seemed to have been open on the language of the Statute, which made patentable “working or making of any manner of new manufactures within this realm”.⁴⁸

Unfortunately for Watt, the Court split evenly on the point, with the result that no judgment was given.⁴⁹ Rooke J saw no problem in allowing patents for methods, since a method “conveys to my understanding the method of a new mode of construction”,⁵⁰ and as such was equivalent to a patent for a machine constructed according to that method. Eyre CJ held that the concept of manufacture was a broad one, which extended not only to things made, but also to the practice of making, which in turn encompassed manners of operating, and new processes producing useful effects.⁵¹ Whilst his Honour did not preclude the patentability of “a method detached from all physical existence whatsoever”,⁵² for any such method to be distinguished from a mere non-patentable principle, it was necessary that it be “embodied and connected with corporeal substances”.⁵³

Heath J took a narrower view to Eyre CJ, holding physicality to be a key feature of his two classes of patentable subject matter – machines and substances. His Honour also considered that a method was non-patentable as lacking in certainty because the patentee “could not specify all the cases, to which his [method] could be applied. The answer seems obvious, that what he cannot specify, he has not invented.”⁵⁴ Buller J arrived at the same result by a slightly different means, holding that a patent was not available for “a method only, without having carried it into effect and produced some new substance”.⁵⁵ His Honour considered that a method was equivalent to a “mode of using a thing, [which] was admitted in the reply not to be a sufficient ground”.⁵⁶

Overall

Broadly speaking, the different approaches of the judges are illustrative in that they capture the uncertainty surrounding the inherent patentability at the time,⁵⁷ but also in that they broadly capture the general approaches which have been taken to determining the boundaries of patentable subject matter ever since.

⁴⁷ Edward C Walterscheid “The Early Evolution of the United States Patent Law: Antecedents (Part 3 continued)” (1995) 77 *Journal of the Patent and Trademark Office Society* 847 at 855.

⁴⁸ *Statute of Monopolies*, s6.

⁴⁹ *Boulton v Bull* (1795) 126 ER 651 at 670.

⁵⁰ *Boulton v Bull* (1795) 126 ER 651 at 659.

⁵¹ *Boulton v Bull* (1795) 126 ER 651 at 666.

⁵² *Boulton v Bull* (1795) 126 ER 651 at 667.

⁵³ *Boulton v Bull* (1795) 126 ER 651 at 667.

⁵⁴ *Boulton v Bull* (1795) 126 ER 651 at 661.

⁵⁵ *Boulton v Bull* (1795) 126 ER 651 at 663.

⁵⁶ *Boulton v Bull* (1795) 126 ER 651 at 663.

⁵⁷ Justine Pila, “Inherent patentability in Anglo-Australian law: a history” (2003) 14 *Australian Intellectual Property Journal* 109 at 117.

After Boulton v Bull

The very same invention was considered again in 1799 in *Hornblower v Boulton*.⁵⁸ In that case, the patentability of the Watt's claims were argued, and in giving judgment Kenyon CJ stated that "having now heard every thing that can be said on the subject, I have no doubt in saying that this is a patent for a manufacture, which I understand to be something made by the hands of man".⁵⁹ In 1819 in *R v Wheeler*,⁶⁰ Abbott CJ expanded on Kenyon CJ's definition, stating that

It is well known that the granting of monopolies was restrained by [the Statute of Monopolies], to the sole working or making of any manufactures, and to the true and first inventor of such manufactures. Now the word "manufactures" has been generally understood to denote either a thing made, which is useful for its own sake, and vendible as such, as a medicine, a stove, a telescope, and many others, or to mean an engine or instrument, or some part of an engine or instrument, to be employed, either in the making of some previously known article, or in some other useful purpose, as a stocking-frame or a steam engine for raising water from mines. Or it may perhaps extend also to a new process to be carried on by known implements, or elements, acting upon known substances, and ultimately producing some other known substance, but producing it in a cheaper or more expeditious manner, or of a better and more useful kind. But no merely philosophical or abstract principle can answer to the word manufacture. Something of a corporeal and substantial nature, something that can be made by man from the matters subjected to his art and skill, or at the least some new mode of employing practically his art and skill, is requisite to satisfy the word.⁶¹

The patentability of principles was revisited in the UK in the 1837 case of *Jupe v Pratt*.⁶² Whilst the non-patentability of principles was upheld, a significant limitation was placed on this exception. Baron Alderson held that "you may take out a patent for a principle coupled with a mode of carrying the principle into effect".⁶³ Further, having developed a mode of carrying the principle into effect, the patentee was "entitled to protect [themselves] from all other modes of carrying the same principle into effect".⁶⁴

Patents as a social contract

Another important change in patent law around this time, which had an indirect influence on patentable subject matter, was a change in the understanding of the nature of the patent grant.

⁵⁸ (1799) 8 TR 99; 101 ER 1285.

⁵⁹ *Hornblower v Boulton* (1799) 101 ER 1285 at 1288. Justice Grose concurred, holding that this was "not a patent for a mere principle, but for the working and making of a new manufacture within the words and meaning of the statute": *Hornblower v Boulton* (1799) 101 ER 1285 at 1289.

⁶⁰ *R v Wheeler* (1819) 106 ER 392 (KB).

⁶¹ *R v Wheeler* (1799) 106 ER 392 at 394-5.

⁶² (1837) 1 Web Pat Cas 145 (Ex).

⁶³ (1837) 1 Web Pat Cas 145 at 147.

⁶⁴ (1837) 1 Web Pat Cas 145 at 147.

Chapter 2: The History of Patentable Subject Matter

Whilst patents were granted in accordance with the Royal Prerogative, they were seen as a privilege rather than a right. All that changed with *Liardet v Johnson*.⁶⁵ In this case, the nature of a patent was characterised not as the conferral of a privilege intended to reward an inventor, but as a social contract wherein the *quid pro quo* of the monopoly grant was that the inventor properly disclose the working of his invention such that the public might gain the benefit of it after the expiry of the monopoly:

The meaning of the Specification is that others may be taught to do a thing for which the patent is granted, & if the specification [is] false, the patent is void for the meaning of the Specification is that after the term [of the Patent] the public shall have the benefit of the discovery.⁶⁶

This theory continues to exert an influence today.⁶⁷

Towards an American Patent System

It was also around this time that a local patent custom began to develop in the US. The relationship between US law and English law was complicated by the fact that the common law did not extend to the US colonies, so patent grants were made in an *ad hoc* manner by colonial legislatures, with the English system having an indirect influence at best.⁶⁸ A national system did not come to be fully realised in the US until the ratification in 1788 of the US Constitution, with its intellectual property clause which granted Congress the power to grant copyright and patent rights “to promote the progress of science and the useful arts”. Perhaps in order to resolve this issue, the second US Patent Act, drafted by Thomas Jefferson in 1793, expressly extended protection to processes. This was achieved by allowing patents on “any new and useful art, machine, manufacture or composition of matter”.⁶⁹ This formulation is the essence of the concept of patentable subject matter in the US to this day.⁷⁰

⁶⁵ *Liardet v Johnson* (1778) 1 Carp Pat Cas 35 (NP).

⁶⁶ James Oldham, *The Mansfield Manuscripts and the Growth of English Law in the Eighteenth Century*, Volume 1 (1992) at 754. Taken from David Brennan, “The Evolution of English Patent Claims as Property Definers” [2005] 4 *Intellectual Property Quarterly* 361; *University of Melbourne Legal Studies Research Paper No. 185* <http://papers.ssrn.com/sol3/papers.cfm?abstract_id=938634> (17 December 2009) at 9.

⁶⁷ The influence of social contract on patent law theory is considered in detail in the next chapter.

⁶⁸ For a detailed discussion see Edward C Walterscheid “The Early Evolution of the United States Patent Law: Antecedents (Part 4)” (1996) 78 *Journal of the Patent and Trademark Office Society* 615. See also Edward C Walterscheid “To Promote the Progress of Useful Arts: American Patent Law and Administration, 1787–1836 (Part 1)” (1997) 79 *Journal of the Patent and Trademark Office Society* 61 at 64. Walterscheid notes that it was “highly unlikely that the Framers were familiar with the common law patent cases that had been decided to 1787, but a majority of them were either lawyers or had some training in the law and thus were knowledgeable about the language of the Statute of Monopolies”: Edward C Walterscheid “To Promote the Progress of Useful Arts: American Patent Law and Administration, 1787–1836 (Part 1)” (1997) 79 *Journal of the Patent and Trademark Office Society* 61 at 70.

⁶⁹ *Patent Act of 1793*, Ch. 11, 1 Stat. 318–323 (US)

⁷⁰ Although the word “art” was replaced with “process”, the language is otherwise unaltered. See *Diamond v Chakrabarty* 447 US 303 (1980) at 308.

Analogous uses

The limits of patentable subject matter were also being explored in attempts to differentiate between non-patentable “analogous uses”, and patentable “novel uses”. In *Losh v Hague*,⁷¹ The way in which this issue was handled was to take a pragmatic view, and resolve the issue not under the banner of inherent patentability, but rather as an issue of novelty:

It would be a very extraordinary thing to say, that because all mankind have been accustomed to eat soup with a spoon, that a man could take out a patent because he says you might eat peas with a spoon. The law on this subject is this: that you cannot have a patent for applying to a well-known thing, which might be applied to 50,000 different purposes, for applying it to an operation which is exactly analogous to what was done before.⁷²

The analogous uses exception was again considered and confirmed in *Harwood v Great Northern Railway*,⁷³ in terms that make clear the link with the *Statute of Monopolies* 1623, s6:

Upon that I think that the law is well and rightly settled, for there would be no end to the interference with trade, and with the liberty of any mechanical contrivance being adapted, if every light difference in the application of a well-known thing were held to constitute a patent ...⁷⁴

The Court did go on to note the difficulty in determining exactly what an analogous use was – a common problem with subject-matter exceptions. It was suggested that whether a use is analogous “always must be a question of degree – a question of more or less – whether the analogy or cognateness of the purposes is so close as to prevent there being an invention”⁷⁵

It will be seen that this interrelationship between novelty and inherent patentability is one which has been echoed in later judgments. *Crane v Price*⁷⁶ considered the interrelationship between newness in the context of a combination of known products. The combination of anthracite and a host air blast in the manufacture of iron was considered patentable on the grounds that it produced “either a new article, or a better article, or a cheaper article to the public, than that produced before by the old method”.⁷⁷ In other words, the newness of the result meant that this was not an analogous use. Similarly, in *Muntz v Foster*⁷⁸ the court held that a “hidden and concealed virtue” in something known before which enabled that known thing to be applied to a useful manufacturing purpose could was patentable.

⁷¹ (1838) 1 Web Pat Cas 202 (NP).

⁷² (1838) 1 Web Pat Cas 202 (NP).

⁷³ (1865) 11 HLC 654; 11 ER 1488.

⁷⁴ *Harwood v Great Northern Railway Company* (1865) 11 ER 1488 at 1499

⁷⁵ *Harwood v Great Northern Railway Company* (1865) 11 ER 1488 at 1493 per Blackburn J.

⁷⁶ (1842) 134 ER 239.

⁷⁷ *Crane v Price* (1842) 134 ER 239 at 410.

⁷⁸ (1843) 2 WPC 93.

Chapter 2: The History of Patentable Subject Matter

Early US case law

Meanwhile, across the Atlantic, US patent law had followed the English lead. In *Le Roy v Tatham*,⁷⁹ the US Supreme Court upheld the non-patentability of principles, stating that “[a] principle, in the abstract, is a fundamental truth; an original cause; a motive; these cannot be patented, as no one can claim in either of them an exclusive right.”⁸⁰ The Court also noted the relationship between invention and discovery, noting that “the invention is not in discovering them, but in applying them to useful objects.”

The next year, in *O’Reilly v Morse*,⁸¹ it was held that an *abstract idea*, apart from its implementation, is not patent-eligible, confirming the majority decision in *Le Roy v Tatham*. Here an over-broad claim which pre-empted all uses of a principle was held to be non-patentable, because of the negative impact on science in the area, and the unjust advantage such a claim would give over future improvements which they did not invent.

The reference to an non-patentable “abstract idea”, although not echoing the exact language of the English cases which held a *principle* to be non-patentable, is effectively the same thing. This is because in both cases there is an absence of a “corporeal substance” or requisite element of physicality. In this way, US jurisprudence on the issue echoes the approaches of Eyre CJ and Heath J in *Boulton v Bull* 100 years earlier.

It is also abstractness which creates the difficulty with process patents. In *Cochrane v Deener*⁸² the Court recognised that process (or method) patents exist at a level of abstraction above machine patents, in that “a process may be patentable, irrespective of the particular form of the instrumentalities used”.⁸³ Purely abstract processes were to properly remain non-patentable however, as a result of the way that the Court defined a process. According to their Honours

[a] process is a mode of treatment of certain materials to produce a given result. It is an act, or a series of acts, performed upon the subject matter to be transformed and reduced to a different state or thing. If new and useful, it is just as patentable as is a piece of machinery. In the language of the patent law, it is an art. The machinery pointed out as suitable to perform the process may or may not be new or patentable; whilst the process itself may be altogether new, and produce an entirely new result. The process requires that certain things should be done with certain substances, and in a certain order; but the tools to be used in doing this may be of secondary consequence.⁸⁴

In other words, a patentable process must involve the transformation of physical subject-matter to a different state or thing.

⁷⁹ 55 US 156 (1852).

⁸⁰ *Le Roy v Tatham* 55 US 156 (1852) at 174-175.

⁸¹ 56 US 62 (1852).

⁸² 94 US 780 (1876).

⁸³ *Cochrane v Deener* 94 US 780 (1876) at 787.

⁸⁴ *Cochrane v Deener* 94 US 780 (1876) at 788.

The modern patent system emerges in the UK

In *Tilghman v Proctor*⁸⁵ the Court affirmed that a patent for a process, irrespective of the particular mode or form of apparatus for carrying it into effect, was admissible under the patent laws of the United States. In particular, the Court recognised the abstract nature of process patents, since processes were “a conception of the mind, seen only by its effects when being executed or performed”.⁸⁶ Thus although an inventor was required to

describe some particular mode or some apparatus by which the process can be applied with at least some beneficial result in order to show that it is capable of being exhibited and performed in actual experience”,⁸⁷

the protection afforded by the resultant patent extended beyond that specific mode of carrying the patent out.

An Australian patent custom?

It was also at around this time that a local Australian patent custom started to develop. Unlike the position in the US, the common law did extend to the Australian colonies, and it was possible for inventors to apply for English patents. However, a local alternative started to develop along US lines, no doubt in order to overcome the difficulties of applying for, and enforcing, long-distance patent protection. Local patents began to be granted by way of private bills in the colonial legislatures, although this was not utilised much.⁸⁸ Subsequent Patent Acts were passed by the various colonial legislatures, modelled on the *Patent Law Act 1852* (UK).⁸⁹ This close tracking of UK law was to continue for some time to come.

4 The modern patent system emerges in the UK

In 1865, in response to growing calls for reform of the patent system,⁹⁰ a Commission was established to review the operation of the patent system. The Commissioners reported that

[i]t is clear that Patents are granted for matter which can hardly be considered as coming within the definition in the Statute of Monopolies, of “a new manufacture”. It is in evidence that the existence of these monopolies embarrasses the trade of a considerable class of persons, artisans, small tradesmen, and others, who cannot afford to fact the expense of litigation, however weak the case against may seem to be; and still a stronger case is made out as to the existence of what may be called

⁸⁵ 102 US 707 (1880).

⁸⁶ *Tilghman v Proctor* 102 US 707 (1880) at 728.

⁸⁷ *Tilghman v Proctor* 102 US 707 (1880) at 729.

⁸⁸ Jeremy Finn, “Particularism versus Uniformity: Factors Shaping the Development of Australasian Intellectual Property Law in the Nineteenth Century” (2000) 5 *Australian Journal of Legal History* 113 at 119-120.

⁸⁹ Jeremy Finn, “Particularism versus Uniformity: Factors Shaping the Development of Australasian Intellectual Property Law in the Nineteenth Century” (2000) 5 *Australian Journal of Legal History* 113 at 120-123.

⁹⁰ Brad Sherman and Lionel Bently, *The Making of Modern Intellectual Property Law: The British Experience 1760-1911* (1999) at 130.

Chapter 2: The History of Patentable Subject Matter

obstructive Patents, and as to the inconvenience caused thereby to manufacturers directly, and through them to the public.⁹¹

The result of this review was the eventual passage of the *Patents, Designs & Trade Marks Act 1883* (UK). Whilst the most important change effected by the system was the move from a system of simple registration to a system of examination,⁹² the new Act impacted upon the approach to patentable subject matter as well. The Act defined an invention as “any manner of new manufacture . . . within section six of the Statute of Monopolies”.⁹³ Whilst the definition itself may not appear to have wrought any change in the law, the practical effect of the definition was to put the focus of the inherent patentability inquiry onto the terms “manner of new manufacture” and “invention”.⁹⁴

In 1884, in the case of *Young v Rosenthal*,⁹⁵ Grove J made what may be the first reference to the non-patentability of mathematics:

An invention of an idea or mathematical principle alone, mathematical formulae or anything of that sort, could not be the subject of a patent. For instance, supposing a person discovered that three angles of a triangle are equal to two right angles, that is an abstract discovery, and would not be the subject of a patent.⁹⁶

On this understanding, it is the abstract nature of mathematics which, like principles and ideas, puts it beyond the realm of patent law. The pairing of the words “abstract” with “discovery” is a pairing which encompassed a change in focus in judicial approach to subject matter which would not be course-corrected until *NRDC*, and even then, not completely.

This contrast of invention with discovery was emphatically approved in *Lane Fox*.⁹⁷ In that case, previous jurisprudence on analogous uses was rationalised by reference to the distinction between a patentable “invention” on the one hand, and an non-patentable “discovery” on the other:

An invention is not the same thing as a discovery. When Volta discovered the effect of an electric current from the battery on a frog’s leg he made a great discovery but no patentable invention. . . A patentee must do something more: he must make some addition, not only to knowledge but to previously known inventions, and he must use his knowledge and ingenuity so as to produce either a new and useful thing or result, or a new method of producing an old thing or result.⁹⁸

⁹¹ United Kingdom, *Report of the Commissioners appointed to the Working of the Law relating to Letters Patent for Inventions* (C5974, 1865) at xi. Cited in Justine Pila, “Inherent patentability in Anglo-Australian law: a history” (2003) 14 *Australian Intellectual Property Journal* 109 at 130-131.

⁹² Klaus Boehm, *The British Patent System, Volume 1: Administration* (1967) at 26-32. Cited in Sam Ricketson & Miranda Richardson, *Intellectual Property: Cases, Materials and Commentary* (3rd ed. 2005) at 660.

⁹³ *Patents, Designs & Trade Marks Act 1883* (UK), s46.

⁹⁴ Justine Pila, “Inherent patentability in Anglo-Australian law: a history” (2003) 14 *Australian Intellectual Property Journal* 109 at 121.

⁹⁵ (1884) 1 RPC 29.

⁹⁶ *Young v Rosenthal* (1884) 1 RPC 29 at 31.

⁹⁷ *Lane Fox v Kensington and Knightsbridge Electric Lighting Company* (1892) 3 Ch. 424 (CA).

⁹⁸ *Lane Fox v Kensington and Knightsbridge Electric Lighting Company* (1892) 3 Ch. 424 per Lindley LJ at 428-429. This approach was echoed in *Reynolds v Herbert Smith & Co Ltd* (1903) 20 RPC 123 at 126 per Buckley J.

Lindley LJ's judgment also evidences a slight change in focus, the import of which may not be immediately obvious. Where earlier cases focused on characterising the invention as abstract versus tangible, the focus here seems to be on knowledge versus application. The latter may have been intended as a mere analogy of the former, but there is nothing inherent in the notion of "application" that requires any physicality. I may "apply" my knowledge of arithmetic to the task of determining my bank balance, but that does not, it is submitted, make such an application suitable for the award of patent rights. More is needed. It is no doubt for this reason that we see the reference to the production of a "thing or result". Again, this may have been intended as a reference to the requirement of some form of tangibility. But the focus on the "result" of a process as determining its patentability has proved to be rife with ambiguity – an ambiguity which has in no small way contributed to the software patent problem.

5 The early 20th Century

United Kingdom

By the turn of the 20th century, the time had come for the patentability requirement to be addressed for a new breed of invention which went beyond the mechanical and chemical inventions of the 19th century, to with ever-more abstract subject matter. In *Cooper's Application*⁹⁹ the applicant sought to patent a particular arrangement of spacing in the layout of the print of a newspaper, "the object being to enable a person reading a newspaper to fold it along these spaces and thus avoid the trouble involved in reading over the folded part of the paper".¹⁰⁰

In considering the alleged invention, the Attorney General noted that a patent could not be granted for a literary composition, which was instead the domain of copyright law.¹⁰¹ Similarly exempted would be an "arrangement of printed matter for its more convenient use from a literary point of view".¹⁰² However, it was held that an alleged invention *could* properly be considered within the domain of patent law if the result or effect of the claims is "a material product of some substantial character".¹⁰³ In the present case, it was held that this was so, since "a newspaper is an article of manufacture. It is an artificial product"¹⁰⁴ and "the alleged utility of his supposed invention [was] purely mechanical".¹⁰⁵ Thus the determinative consideration, as with earlier cases, continued to be the existence of some physical or corporeal subject matter. The case does however rest on a very fine distinction, in that it illustrates that the type of

⁹⁹ *Re Cooper's Application for a Patent* (1901) 19 RPC 53 (AG)

¹⁰⁰ *Re Cooper's Application for a Patent* (1901) 19 RPC 53 (AG) at 53.

¹⁰¹ *Re Cooper's Application for a Patent* (1901) 19 RPC 53 (AG) at 54.

¹⁰² *Re Cooper's Application for a Patent* (1901) 19 RPC 53 (AG) at 54. Also see 54-55 where the Attorney-General further explains that such a "production of a literary kind [is one which] may enable the reader more readily to appreciate the sense of the author". See also *Virginia-Carolina Chemical Corp's Application* [1958] RPC 35, that the intellectual or visual content relates to the fine and not the useful arts; *Pitman's Application* [1969] RPC 646 that any matter having a purely intellectual, literary or artistic connotation is not patentable. For further examples see Australian Patent Office, "2.9.2.8 Printed Matter" in *Manual of Practice and Procedure* (2006) <http://www.ipaustralia.gov.au/pdfs/patentsmanual/WebHelp/Patent_Examiners_Manual.htm> (accessed 9 February 2010).

¹⁰³ *Re Cooper's Application for a Patent* (1901) 19 RPC 53 (AG) at 54.

¹⁰⁴ *Re Cooper's Application for a Patent* (1901) 19 RPC 53 (AG) at 55.

¹⁰⁵ *Re Cooper's Application for a Patent* (1901) 19 RPC 53 (AG) at 54.

Chapter 2: The History of Patentable Subject Matter

protection available for the arrangement of words on a page will depend on the purpose for which such arrangement is undertaken. For that reason, characterisation of the invention in the present case was particularly important:

[T]he mere fact that one feature in the scheme was a printed sheet, or coupon, or ticket, or any equivalent, the invention being alleged to be an arrangement of words upon the sheet, would not form a subject in respect to which a Patent might properly be granted, because it would not be manufacture; in substance it would be a scheme, and the mere arrangement of printed matter in connection with such scheme would not prevent you from *looking at the real object of the Applicant* (emphasis added)¹⁰⁶

Non-patentability of principles revisited

In *Hickton's Patent Syndicate*,¹⁰⁷ the court considered an appeal of an action for infringement in which the plaintiff's patent was found invalid for want of subject matter. At the trial, Swinfen-Eady J considered the relationship between idea and invention, and suggested that "[a]n idea may be new and original and very meritorious, but unless there is some invention necessary for putting the idea into practice it is not patentable".¹⁰⁸ This approach draws a sharp line between an idea and its implementation, requiring that the implementation itself conform to the requirements of a manner of manufacture. However, the Court of Appeal disagreed, holding that "invention may lie in the idea, and it may lie in the way in which it is carried out, and it may lie in the combination of the two; but if there is invention in the idea plus the way of carrying it out, then it is good subject matter for Letters Patent."¹⁰⁹

Hickton's case evidences a bold new leap forward into the award of monopolies over abstract subject matter. In fact, the case law of this era is hard to digest, and even more difficult to reconcile. The non-patentability of ideas is a case in point. It has been set out in detail above how principles and abstract ideas are not patentable. Yet on reading the case law from this era, it becomes clear that whilst an idea may not *of itself* be claimed, once it is coupled with an application, it is not considered problematic that the monopoly gives effective protection to the idea itself.¹¹⁰ As Buckley LJ stated in *Hickton's* "I think you can have a Patent for an idea, which is new and original and very meritorious, if you suggest a way of carrying it out."¹¹¹

Where the invention lies in the idea, this may not be difficult. As the Court noted, "once the idea of applying some well known thing for a special and new purpose is stated it may be very obvious how to give effect to that idea".¹¹² Thus there may be little upon which to differentiate

¹⁰⁶ *Re Cooper's Application for a Patent* (1901) 19 RPC 53 (AG) at 54.

¹⁰⁷ *Hickton's Patent Syndicate v Patents and Machine Improvements Co. Ltd.* (1909) 26 RPC 339.

¹⁰⁸ Taken from *Hickton's Patent Syndicate v Patents and Machine Improvements Co. Ltd.* (1909) 26 RPC 339 at 347. It should be noted that such an argument would today be dealt with under the banner of inventive step, but at the time, it was not a separate inquiry.

¹⁰⁹ *Hickton's Patent Syndicate v Patents and Machine Improvements Co. Ltd.* (1909) 26 RPC 339 at 348 per Fletcher-Moulton LJ. See also 347 per Cozens-Hardy MR; 348 per Buckley LJ.

¹¹⁰ With one important caveat, which will be discussed shortly.

¹¹¹ *Hickton's Patent Syndicate v Patents and Machine Improvements Co. Ltd.* (1909) 26 RPC 339 at 348.

¹¹² *Hickton's Patent Syndicate v Patents and Machine Improvements Co. Ltd.* (1909) 26 RPC 339 at 348 per Buckley LJ.

a non-patentable new principle in the abstract, from a patentable new-principle-plus-obvious-application. Fine distinctions such as this can be difficult to grasp, and it is understandable that Swinfen-Eady J was motivated to draw a brighter line between patentable and non-patentable, through a requirement that the application of an idea itself be inventive. However, the Court of Appeal, in rejecting this distinction, noted that the practical effect of such a requirement would be to “deprive of their reward a very large number of meritorious inventions that have been made”.¹¹³

Though the distinction may be fine, it would be wrong to say that demonstrating an application amounts to a requirement of form rather than substance. In considering an appeal from the Supreme Court of Canada, Viscount Cave in *Permutit Co v Borrowman*¹¹⁴ described the minimum amount of application required as follows: “[i]t is not enough for a man to say that an idea floated through his brain; he must at least have reduced it to a definite and practical shape before he can be said to have invented a process.”¹¹⁵ What amounts to a definite and practical shape will be a matter to be considered in individual cases.

The Court’s approach in *Hickton’s case* also highlights the importance of identifying where the “newness” in the claimed invention lies. This is because

... if the principle is new, and you show one mode of carrying it into effect, you may protect yourself against all other modes of carrying the principle into effect. If however, the principle is not new, you can only protect yourself against those modes of carrying it into effect which are substantially the same as the mode you have yourself invented, the question being in each case what is the pith and marrow of the invention sought to be protected, and it being impossible to treat any principle already known as part of such pith and marrow. Thus, where the principle is old, a claim to all modes of carrying it into effect will avoid the Patent. ...¹¹⁶

As inventions became more abstract, drawing ever finer distinctions between idea and application can lead to the spectre of over-protection. This is because patents provide protection at a level of abstraction above the application actually described, extending to equivalents.¹¹⁷ One solution, espoused by the Court in *British United Shoe Machinery Co. Ltd. v. Simon Collier Ltd*¹¹⁸ was to look at the practical effect of awarding the patent. As Fletcher-Moulton LJ noted:

... I cannot find any authority for the proposition that, if the principle had been new, the inventor could have protected himself against all means of solving the

¹¹³ *Hickton’s Patent Syndicate v Patents and Machine Improvements Co. Ltd.* (1909) 26 RPC 339 at 348 per Fletcher-Moulton LJ. His Honour drew on the example of *Boulton v Watt* above in illustrating this concern.

¹¹⁴ (1926) 43 RPC 356.

¹¹⁵ *Permutit Co v Borrowman* (1926) 43 RPC 356

¹¹⁶ *British United Shoe Machinery Co. Ltd. v. Simon Collier Ltd* (1909) 26 RPC 21 at 49.

¹¹⁷ *RCA Photophone Ltd v Gaumont-British Picture Corporation* (1936) 53 RPC 167 at 197 per Romer LJ. See also *British United Shoe Machinery Co. Ltd. v. Simon Collier Ltd* (1909) 26 RPC 21 at 49 (discussed below); *David Kahn Inc v Conway Stewart & Co Ltd* [1974] RPC 279 “A patentee may rightly claim a monopoly wider in extent than what he had invented” at 319 per BW Budd QC. This was also recognised by the US Supreme Court in *Cochrane v Deener* (1876) 94 US 780 at 787 (discussed above at 48).

¹¹⁸ (1909) 26 RPC 21.

Chapter 2: The History of Patentable Subject Matter

problem.¹¹⁹

The requirement of a specific application also restricts the patentability of abstract matter. The language of the claims must be drafted so as to define the problem sought to be solved, and avoid any language which may be read as extending to such abstract matter. In other words it would be difficult to draft claims in such broad terms as to protect against all uses of a principle, whilst still practically defining an invention in narrow enough terms that it meets the disclosure requirement. This point was captured by the Court in *British United Shoe Machinery*:

... a Claim ought to state, either expressly or by plain reference, what is the invention for which protection is demanded. The very object of a Claim is to inform others of the limits of such protection. ... If an inventor has, in fact, made a Claim which, *because of its width, is not proper subject matter*, his Patent must be invalid...¹²⁰

As Lord Wright MR put it in *RCA Photophone*, “the precise ambit of the claim must depend on the language used.”¹²¹ Such an approach is consistent with the social contract theory of patent law advanced in *Liardet v Johnson*.¹²² This is evident from the judgment of Romer LJ in *RCA Photophone* who noted the importance of delimiting the boundaries of the monopoly:

It is the duty of a patentee by his claim to make quite clear what is the ambit of his monopoly in order that works in the art may be left in no doubt as to the territory that is forbidden them during the life of the patent. If he fails to do this, his patent becomes a public nuisance. It is equally incumbent upon him to describe at least one way, and the best way known to him, of carrying his invention into effect, in order that, when his monopoly comes to an end, the workers in the art may turn the invention to account. This is the consideration he pays for his monopoly.¹²³

Romer LJ also briefly touched upon a final way in which the non-patentability of principles is guaranteed (although it may be of little comfort to a patentee’s competitors):

The Plaintiffs do not, of course, contend that a man can under any circumstances obtain a monopoly in an idea or principle, however new it may be. The discoverer of an idea or a principle obviously cannot prevent the rest of the world from entertaining the idea or asserting the principle...¹²⁴

Put another way, the entertaining of the idea or assertion of the principle does not amount to infringement. Any attempt to formulate a claim in such a manner would have to amount to non-patentable subject matter.¹²⁵

¹¹⁹ Note the (unhelpful) caveat put on this contention in *David Kahn Inc v Conway Stewart & Co Ltd (No 2)* [1974] RPC 279 at 319-320: “He cannot claim all solutions to a problem *unless invention lies in identification of the problem*” (emphasis added).

¹²⁰ *British United Shoe Machinery Co. Ltd. v. Simon Collier Ltd* (1909) 26 RPC 21 at 50 (emphasis added).

¹²¹ *R.C.A. Photophone Ltd. v. Gaumont-British Picture Corporation* (1936) 53 RPC 167 at 186-187.

¹²² Discussed at page 45 above.

¹²³ *R.C.A. Photophone Ltd. v. Gaumont-British Picture Corporation* (1936) 53 RPC 167 at 194.

¹²⁴ *R.C.A. Photophone Ltd. v. Gaumont-British Picture Corporation* (1936) 53 RPC 167 at 195.

¹²⁵ There is an obvious overlap here with the later US doctrine of “mental steps”, discussed in Chapter 3.

One possible alternative interpretation of these cases may be that the term ‘principle’ has assumed a truly unfortunate double meaning within patent law. On the one hand, we have an “abstract” principle or idea, which remains non-patentable. On the other, we have what is being referred to in the instant case, a “mechanical” principle – the physical manner in which the abstract principle or idea is carried into effect. Support for this interpretation can for instance be found in *British Shoe Machinery*:

In these cases ... I think the word ‘principle’ is used to denote some chemical or mechanical principle which the inventor employs in solving some problem, the solution of which is the object of his invention, and does not refer to the problem itself.¹²⁶

Similarly, in *R.C.A. Photophone Ltd. v. Gaumont-British Picture Corporation*¹²⁷ Lord Wright MR noted that what “is often, very inaptly, called a principle, ... [when what] is meant [is] some general method of manufacture”.¹²⁸ Lord Wright’s approach to determining the issue of infringement clearly contemplates a more limited notion of a principle as being something which is tied to the problem sought to be solved by its application:

In the present case, both the Appellants and the Respondents have sought to solve a known problem, namely, how to eliminate ground noises. In my opinion, the Respondents’ method ... is not in any sense a replica of the Appellants’. It seeks to solve the problem on different lines...¹²⁹

It is clear from this statement that it is the “lines” along which a problem is sought to be solved, rather than the solution to the problem, which is patentable invention. Thus one must be very careful in determining the scope of the patent by reference to its result:

In this case it might be said that the result achieved was the same ... But I think that involves an incorrect use of the word “result”, which in a case like this should be used in the narrower sense of what is practically achieved in physical fact; “result” is properly used in this connection as referring, not to the end, but to the means.¹³⁰

A focus on the problem to be solved may thus solve some of the difficulties associated with reconciling claims to abstract principles through claims to specific applications. This is because they limit the scope of a patent claim to the context of the problem to be solved, and thereby reduce the likelihood of broad claims to general principles being recognised.

¹²⁶ *British United Shoe Machinery Co. Ltd. v. Simon Collier Ltd* (1909) 26 RPC 21 at 49.

¹²⁷ (1936) 53 RPC 167.

¹²⁸ *R.C.A. Photophone Ltd. v Gaumont-British Picture Corporation* (1936) 53 RPC 167 at 186.

¹²⁹ *R.C.A. Photophone Ltd. v Gaumont-British Picture Corporation* (1936) 53 RPC 167 at 190.

¹³⁰ *R.C.A. Photophone Ltd. v Gaumont-British Picture Corporation* (1936) 53 RPC 167 at 191.

Chapter 2: The History of Patentable Subject Matter

Commerce and trade

It was also at around this time that a reconsideration of the notion of manufacture lead to a different limit on patentability being espoused. The court in *C & W's Application*,¹³¹ in rejecting the patentability of methods of medical treatment, did not look directly to a physicality requirement, but instead held that patentable subject matter must be “in some way associated with commerce and trade”.¹³² In particular, the Solicitor interpreted the phrase “manner of new manufacture” in the context of s6 of the *Statute of Monopolies* as requiring an association “with the manufacture or sale of commercial products”, beyond that “a thing newly made, or a substance which, if made before is improved in its nature; or ... a machine or a process that can be used in making something that is, or may be of commercial value”.¹³³ It should be noted that the Solicitor-General’s use of the terms “thing”, “substance” and “something” indicate however that the physicality requirement was still present in his thinking. In any event, the immediate effect of this requirement was to preclude from patentability any methods of medical treatment, on the basis that the use of such methods, although improving the person on which they operated, were nonetheless not “a process employed in any form of manufacture or of trade”.¹³⁴

It is also worth passing mention at this point that the Solicitor-General was quick to reject the morality of the alleged invention as a relevant consideration, holding that “the question of humanity ought not to affect the decision in a case such as this”,¹³⁵ and finding refuge instead in “arriving at a conclusion upon the terms of the Section of the Act of Parliament”.¹³⁶ This is but an early example of the claimed moral neutrality of patent law, a point which will be taken up later in this thesis.

Morton's rules

The physicality requirement of *Coopers Application* and the commerciality requirement of *C&W's Application* were given a joint formulation in the case of *Re GEC*¹³⁷ wherein it was held that to be patentable an invention had to involve the production, improvement or preservation of “some vendible product”.¹³⁸ In argument, counsel asserted that:

a method or process is a manner of manufacture if it (a) results in the production of some vendible product or (b) improves or restores to its former condition a vendible product or (c) has the effect of preserving from deterioration some vendible product to which it is applied.¹³⁹

¹³¹ *Re C & W's Application for a Patent* (1914) 31 RPC 235 (SG).

¹³² *Re C & W's Application for a Patent* (1914) 31 RPC 235 at 235.

¹³³ *Re C & W's Application for a Patent* (1914) 31 RPC 235 at 235-236.

¹³⁴ *Re C & W's Application for a Patent* (1914) 31 RPC 235 at 236.

¹³⁵ *Re C & W's Application for a Patent* (1914) 31 RPC 235 at 236.

¹³⁶ *Re C & W's Application for a Patent* (1914) 31 RPC 235 at 236.

¹³⁷ *Re GEC's Application* (1941) 60 RPC 1.

¹³⁸ *Re GEC's Application* (1941) 60 RPC 1 at 4.

¹³⁹ *Re GEC's Application* (1941) 60 RPC 1 at 4.

Morton J gave this formulation limited support, as follows:

In saying this I am not attempting to cover every case which may arise by a hard and fast rule. At the moment I cannot think of any method or process which does not come within (a), (b) or (c) and yet is a manner of manufacture. Nor can I think, at the moment, of any method or process which does come within (a), (b) or (c) and yet is not a manner of manufacture. I do not intend, however, to rule out the possibility that such methods or processes may be brought to my attention hereafter. Nor do I intend to rule out the possibility that I may hereafter be convinced by argument that the rule which I have just expressed ought to be modified...¹⁴⁰

Despite Morton J's equivocation, his remarks were adopted in a number of subsequent cases, and for nearly two decades assumed the mantle of a test. What is worse, this test was applied in an inconsistent fashion.¹⁴¹ For example, in finding an electrical oscillation to fall within the concept of a vendible product in *Rantzen*,¹⁴² Evershed J suggested that the purpose of the vendible product concept was merely to lay "proper emphasis upon the trading or industrial character of the processes intended to be comprehended by the Acts".¹⁴³ However, other cases had used the absence of a vendible product to exclude

two main types of process from 1942 to 1959. The first type involved the production or treatment of ephemeral matter, and the second involved the treatment of soil to improve its cropbearing utility; the latter on the ground that the only vendible product was the crops nourished by the soil which were not affected by the treatment itself.¹⁴⁴

For example, in *Re Lenard's Application*,¹⁴⁵ this requirement of "industrial or commercial or trading character"¹⁴⁶ derived from the vendibility requirement in Morton's rules,¹⁴⁷ was held to preclude from patentability "a method of agricultural or horticultural treatment".¹⁴⁸ In fact, Pila notes that the result of the adoption of the vendible product test was the exclusion from patentability of "three emergent subject matter of the 20th century: surgical and therapeutic methods; agricultural, horticultural and other biotechnological subject matter; and informational products such as schemes and aesthetic works."¹⁴⁹ Thus it was only a matter of time before it would be swept aside.

¹⁴⁰ *Re GEC's Application* (1941) 60 RPC 1 at 4.

¹⁴¹ Ann L. Monotti, "The Scope of Manner of Manufacture Under the Patents Act 1990 (Cth) after *Grant v Commissioner of Patents*," (2006) 34(3) *Federal Law Review* 461 at 463.

¹⁴² *Re Application by Rantzen* (1946) 64 RPC 63.

¹⁴³ *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 271.

¹⁴⁴ Justine Pila, "Inherent patentability in Anglo-Australian law: a history" (2003) 14 *Australian Intellectual Property Journal* 109 at 141.

¹⁴⁵ *Re an Application for a Patent by Lenard* (1954) 71 RPC 190.

¹⁴⁶ *Re an Application for a Patent by Lenard* (1954) 71 RPC 190 at 192 per Lloyd-Jacob LJ.

¹⁴⁷ *Re an Application for a Patent by Lenard* (1954) 71 RPC 190 at 191-192 per Lloyd-Jacob LJ.

¹⁴⁸ *Re an Application for a Patent by Lenard* (1954) 71 RPC 190 at 192 per Lloyd-Jacob LJ.

¹⁴⁹ Justine Pila, "Inherent patentability in Anglo-Australian law: a history," (2003) 14 *Australian Intellectual Property Journal* 109 at 143-144.

Chapter 2: The History of Patentable Subject Matter

In *Re Elton and Leda Chemicals Ltd's Application*¹⁵⁰ Lloyd-Jacob J considered the patentability of a method of dispersing fog by introducing a surface-active agent to coalesce the droplets into raindrops. His Honour noted the convenience of the 'vendible product' test, and noted that in the instant case, if applied with "a little latitude it might afford some assistance in the present case, for a fog-free atmosphere or a deliberately induced rainfall could be a factor in the site value of the land whereon the Applicants' process was applied".¹⁵¹ This indicates a willingness on the part of his Honour to go beyond a strict, literal interpretation of both what is vendible, and what constitutes a product in the application of Morton J's "rules", much as Evershed J had done in earlier cases.

In relation to the meaning of the term "product", his Honour went on to say:

[t]here has been no question, at any rate since before the year 1800, that the expression 'manner of manufacture' in the [*Statute of Monopolies*] must be construed in the sense of including a practice of making as well as the means of making and the product of making. It has thus been appreciated that, although an inventor may use no newly devised mechanism, nor produce a new substance, none the less he may, by providing some new and useful effect, appropriate for himself a patent monopoly in such improved result by covering the mode or manner by means of which his result is secured. Seeing that the promise which he offers is some new and useful effect, there must of necessity be some product whereby the validity of his promise can be tested.¹⁵²

Lloyd-Jacob J clearly found some latitude in what might amount to a "product" – no longer does such a product need to be a mechanism or substance. But this does not do away with a physical embodiment entirely. What is sought to be appropriated is some "effect" or its synonym, a "result". When considering what is meant by these terms, it is apposite to keep in mind what was said in *RCA Photophone* about results, namely "what is practically achieved in physical fact: referring not to the end, but to the means."¹⁵³ These terms, by their common usage tend to suggest that the important factor is the *end* result, the effect as opposed to the cause. But this is clearly not what was meant. Lloyd-Jacob J himself notes that the patent secured covers "the mode or manner by means of which [the] result is secured".¹⁵⁴ That this is the correct interpretation is also apparent from earlier cases which make it clear that a claim to all solutions to a problem is so broad as to not constitute the proper subject of a patent.¹⁵⁵ Through this broader understanding of a product then, continues the requirement of physicality developed over the cases thus far described.

¹⁵⁰ (1957) RPC 267.

¹⁵¹ *Elton and Leda Chemicals Ltd's Application* [1957] RPC 267 at 269.

¹⁵² *Re Elton and Leda Chemicals Ltd's Application* [1957] RPC 267 at 268-269.

¹⁵³ *R.C.A. Photophone Ltd v Gaumont-British Picture Corporation* (1936) 53 RPC 167 at 191.

¹⁵⁴ *Re Elton and Leda Chemicals Ltd's Application* [1957] RPC 267 at 268-269.

¹⁵⁵ In particular see the discussion of *British United Shoe Machinery Co Ltd v Simon Collier Ltd* (1909) 26 RPC 1 at 49 above.

United States

Cases in the US at around this time had begun to expand on the notion of non-patentability of abstract subject matter. In *Mackay Radio v RCA*¹⁵⁶ the Supreme Court held that “while a scientific truth, or the mathematical expression of is not patentable invention, a novel and useful structure created with the aid of knowledge of scientific truth may be”¹⁵⁷ Similarly, in rejecting an improved combination of fire-fighting apparatus, the US Court of Customs and Patent Appeals in *Patton*¹⁵⁸ noted that

a system of transacting business, apart from the means for carrying out such system, is not within the purview of [the section defining patentable subject matter], ... nor is an abstract idea or theory, regardless of its importance or the ingenuity with which it was conceived, apart from the means for carrying such idea or theory into effect, patentable subject matter.

Whilst these cases ostensibly suggest explicit exceptions for scientific truths, mathematics, and business methods, it is important to note both the association of such methods with abstract ideas, and also the reinforcement of the notion that an application thereof might be patentable. What this suggests is that the theory of patent law had not changed, and the subject matter specifically recounted is more of an example of the types of abstract principles held non-patentable in earlier cases.

The increasingly abstract nature of inventions was, in the US, finding Courts willing to express the boundary line in different ways. In *Middleton*¹⁵⁹ the CCPA held that “a process which is the mere function of an apparatus is not patentable, and a process which cannot be carried out apart from a particular apparatus is the mere function of that apparatus”.¹⁶⁰ The 1948 case of *Funk Bros Seed Co v Kalo Inoculant Co*¹⁶¹ suggested a reason for the non-patentability of principles, namely that they were “part of the storehouse of knowledge of all men . . . free to all men and reserved exclusively to none”¹⁶²

Australia

Before the turn of the Century, the various Australian colonies had all passed patent laws based on the 1883 UK Act.¹⁶³ However, after Federation, in 1903, Australian patent law was centralised, with the various States handing administration of their Patent Acts to the Commonwealth. The 1903 Act was also based on the 1883 UK Act,¹⁶⁴ thereby incorporating all previous UK jurisprudence on patentability. This reliance on UK jurisprudence was to continue, largely

¹⁵⁶ (1939) 306 US 86.

¹⁵⁷ *Mackay Radio v RCA* (1939) 306 US 86 at 94.

¹⁵⁸ (1942) 29 CCPA 982; 127 F.2d 324.

¹⁵⁹ *In re Middleton* (1948) 167 F.2d 1012; 77 U.S.P.Q. (BNA) 615.

¹⁶⁰ *In re Middleton* 167 F.2d 1012; 77 U.S.P.Q. (BNA) 615 (1948) at 616.

¹⁶¹ *Funk Bros Seed Co v Kalo Inoculant Co* 333 US 127 (1948).

¹⁶² *Funk Bros Seed Co v Kalo Inoculant Co* 333 US 127 (1948) at 130.

¹⁶³ HN Walker, *Australian Patents: The Australian Patent System Explained* (1949) at xxiii.

¹⁶⁴ HN Walker, *Australian Patents: The Australian Patent System Explained* (1949) at xxiii.

Chapter 2: The History of Patentable Subject Matter

unabated until the UK adopted the European Patent Convention in 1977. The 1903 Act itself remained unchanged for almost 50 years, despite two reviews in 1935¹⁶⁵ and 1950.¹⁶⁶ Even then, the changes introduced in the 1952 Act were minor, and deliberately tracked UK law.¹⁶⁷

In *Rogers v Commissioner of Patents*,¹⁶⁸ the High Court handed down judgment in its first patent case, rejecting a patent for a method of burning standing timber. Griffiths CJ adopted the approach of Lindley LJ in *Lane Fox*, holding that the new use of a known object will be patentable only where the novelty lies in the mode of using it, as distinguished from novelty of purpose. His Honour held that the present invention was at best “a direction as to how to apply the law of gravitation in order to keep in contact two bodies, one of which is being consumed”.¹⁶⁹ O’Connor J expressly referred to *Wheeler’s case* in holding that a new method of producing an old result, to be patentable “must either produce some vendible article or must be carried out by some mechanical contrivance or some substance the use or adaptation of which for the purpose of working the new method is part of the invention.”¹⁷⁰ Isaacs J strongly dissented, finding the approach of the majority to be too narrow. His Honour, relying on the conception of patentability espoused by Fletcher Moulton LJ in *Hickton’s case*, would have allowed the patent on the basis that it was “the combination of the idea or principle, of machine or apparatus, and of system or *modus operandi*, which constitute[d] the total process the subject of the application.”¹⁷¹

In *Maeder v Busch*,¹⁷² the a patent for an improved process for producing permanent waves in human hair was ultimately rejected on the grounds of prior public use. However, in *obiter* Dixon J discussed whether the subject matter in question was patentable in some detail. His Honour noted that “[t]o be patentable an invention must relate to an art. ... But the ultimate end in view is the production or treatment of, or effect upon, some entity.”¹⁷³ His Honour doubted that the claimed process could be patentable, as

the object [was] not to produce or aid the production of any article of commerce. No substance or thing forming a possible subject of commerce or a contribution to the productive arts is to be brought into existence by means of or with the aid of the process.¹⁷⁴

Dixon J did not close the door completely however, noting that “a widening conception of a manner of new manufacture has been a characteristic of the growth of patent law.”¹⁷⁵ Similarly,

¹⁶⁵ The Knowles Committee review was appointed in 1935. The report of the Committee is annexed to the report of the later Dean Committee (below).

¹⁶⁶ The Dean Committee were appointed in 1950, and delivered their final report in 1952. See *Report of the Committee appointed by the Attorney-General of the Commonwealth to Consider what Alterations are Desirable in the Patent Law of the Commonwealth* (the Dean Committee), 1952.

¹⁶⁷ See Sam Ricketson, “The Future Role of Australian Intellectual Property Law Reform and Administration” (1992) 3(1) *Australian Intellectual Property Journal* 3 at 12.

¹⁶⁸ (1910) 10 CLR 701.

¹⁶⁹ *Rogers v Commonwealth* (1910) 10 CLR 701 at 708.

¹⁷⁰ *Rogers v Commonwealth* (1910) 10 CLR 701 at 712.

¹⁷¹ *Rogers v Commonwealth* (1910) 10 CLR 701 at 718.

¹⁷² (1938) 59 CLR 684.

¹⁷³ *Maeder v Busch* (1938) 59 CLR 684 at 705.

¹⁷⁴ *Maeder v Busch* (1938) 59 CLR 684 at 706.

¹⁷⁵ *Maeder v Busch* (1938) 59 CLR 684 at 706.

McTiernan J held that the process could conceivably “be good subject matter of a patent ... upon a wide, and perhaps novel, interpretation of the words ‘manner of manufacture.’”¹⁷⁶

*Microcell*¹⁷⁷ revisited the non-patentability of analogous uses, holding that

[I]t is not an inventive idea for which a monopoly can be claimed to take a substance which is known and used for the making of various articles, and make out of it an article for which its known properties make it suitable, although it has not in fact been used to make that article before.¹⁷⁸

The need to demonstrate some form of newness other than a novelty of use was again reiterated in *Griffin v Isaacs*¹⁷⁹ where Dixon J held “that if a subject matter when considered as a working invention were indistinguishable in merit or principle from an existing invention, it would be incapable of possessing the newness or threshold inventiveness required of manners of new manufacture and unable to support a patent on that basis”.¹⁸⁰ In other words, the concept of mechanical inventiveness discussed in *Coopers case* is required.

6 The NRDC case

The foundation of the current approach to the patentable subject matter issue in Australia was set in 1959, in the landmark case of *NRDC*.¹⁸¹ In this case it was held that the proper approach to determining patentable subject matter was not to look for a convenient formula, but to ask the question “is this a proper subject of letters patent according to the principles which have been developed for the application of section 6 of the Statute of Monopolies?”¹⁸² Despite the apparently broad scope which such a question suggests, typical summaries of this case have refined its content along the lines that to be patentable a process “requires a mode or manner of achieving an end result which is an artificially created state of affairs of utility in the field of economic endeavour.”¹⁸³ Subsequent cases have confirmed this so-called “test” as the primary test for patentability in the Australian jurisdiction.¹⁸⁴

Whilst some have applauded the flexibility of this definition,¹⁸⁵ trying to explain its exact meaning by reference to this shorthand is a difficult task, which either leads back along a circular path to what has come before, or on to other similarly nebulous explanations. In the end, most discussions of this test tend to devolve quickly into the specifics of whether particular

¹⁷⁶ *Maeder v Busch* (1938) 59 CLR 684 at 708.

¹⁷⁷ *Commissioner of Patents v Microcell Ltd* (1959) 102 CLR 232.

¹⁷⁸ *Commissioner of Patents v Microcell Ltd* (1959) 102 CLR 232 at 249; see also 251.

¹⁷⁹ (1938) 12 ALJ 169.

¹⁸⁰ Justine Pila, “Inherent patentability in Anglo-Australian law: a history,” (2003) 14 *Australian Intellectual Property Journal* 109 at 137.

¹⁸¹ *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252.

¹⁸² *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 269.

¹⁸³ *CCOM Pty Ltd v Jiejing Pty Ltd* (1996) 21 FCR 260 at 295. See also *Welcome Realtime v Catuity* (2001) 113 FCR 110 at 137: “In my opinion the Patent does produce an artificial state of affairs ... Moreover this result is beneficial in a field of economic endeavour. ...”.

¹⁸⁴ See for example *Joos v Commissioner of Patents* (1972) 126 CLR 611 at 616 per Barwick CJ; *CCOM Pty Ltd v Jiejing Pty Ltd* (1996) 21 FCR 260 at 287.

¹⁸⁵ Industrial Property Advisory Council, *Patents, Innovation and Competition in Australia* (1984) at 41.

Chapter 2: The History of Patentable Subject Matter

categories should be patentable or not.¹⁸⁶ On this basis it is little wonder then that Christie has suggested that so-called *NRDC* test is of “such a wide, elastic and amorphous character as to cover almost all newly-created subject matters or processes.”¹⁸⁷ Rather than being a watershed, Christie submits that a better description would be a “bombshell”.¹⁸⁸ Van Caenegem similar criticises the test as “vague and embryonic”.¹⁸⁹

It will be demonstrated that this reductionist account of the case is misleading, as the Court were at great pains to avoid reducing the issue of inherent patentability to any particular formula. The emphasis of the case was on keeping an eye on the past when looking to the future.

The claimed invention in this case was a method of using a chemical substance for eradicating weeds from crop areas without damaging desired growth. The report of the examiner contained objections that the claims were directed to “the mere use of known substances – which use also does not result in any vendible product”.¹⁹⁰ The Deputy Commissioner then directed that the first three claims in the application be removed, on the basis that the method claimed “did not result in any vendible product.”¹⁹¹ The applicant appealed directly to the High Court.

Analogous uses

The Court first noted the power of the Commissioner to reject claims for a process which does not amount to a manufacture because it is “nothing but a new use of a known substance”,¹⁹² or in other words an analogous use. Such uses were said to be non-patentable “[b]ecause in the postulated state of knowledge the new purpose is no more than analogous to the purposes for which the utility of the substance is already known; and therefore your suggestion of the new purpose lacks the quality of inventiveness.”¹⁹³ However, the Court held that the requisite inventiveness did exist where the new use “consists in taking advantage of a hitherto unknown or unsuspected property of the material”,¹⁹⁴ and was to be found “in the suggestion that the substance may be used to serve the new purpose”.¹⁹⁵ That being the case,

provided that a practical method of so using it is disclosed and that the process comes within the concept of [a manner of manufacture]... all the elements of a

¹⁸⁶ See for example Sam Ricketson and Miranda Richardson, *Intellectual Property: Cases, Materials and Commentary* (3rd ed. 2005); Jill McKeough, Andrew Stewart & Philip Griffith, *Intellectual Property in Australia* (3rd ed. 2004); Rocque Reynolds & Natalie P Stoianoff, *Intellectual Property: Text and Essential Cases* (2nd ed. 2005).

¹⁸⁷ Jack Starke, “The Patenting of Animal Forms With New Traits,” (1987) 61 *Australian Law Journal* 324 at 325-6. Andrew Christie goes further, suggesting that the result of *NRDC* is that the patentable subject matter test has been ‘annihilated’. See Andrew Christie, “Some Observations on the Requirement of Inherent Patentability in the Context of Business Method Patents,” (2000) 41 *IP Forum* 16 at 17.

¹⁸⁸ Andrew Christie, “Some Observations on the Requirement of Inherent Patentability in the Context of Business Method Patents” (2000) 41 *IP Forum* 16, at 13.

¹⁸⁹ William van Caenegem, “The Technicality Requirement, Patent Scope and Patentable Subject Matter in Australia” (2002) 13 *Australian Intellectual Property Journal* 41, at 44.

¹⁹⁰ *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 261 per examiner.

¹⁹¹ *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 261.

¹⁹² In support of that proposition, the Court recited the wording of Lord Buckmaster in *Re BA's Application* (1915) 32 RPC 348 at 349. See *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 262.

¹⁹³ *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 262, citing in support *Elias v Grovesend Tinsplate Co* (1890) 7 RPC 455 at 468 per Bowen LJ.

¹⁹⁴ *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 262.

¹⁹⁵ *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 262.

patentable invention are present: see the *Microcell* case. It is not necessary that in addition the proposed method should itself be novel or involve any inventive step: *Hickton's* [case].¹⁹⁶

Inventiveness; discovery v invention; knowledge v application

The Court espoused that such an approach was consistent with the view of the Court in *Lane Fox*, “that a man who discovers that a known machine ... can produce effects which no one before him knew could be produced by it has made a discovery, but has not made a patentable invention”¹⁹⁷ The Court then went on to further emphasise the distinction, noting that the difference lay in an inventor using their “knowledge and ingenuity as to produce either a new and useful thing or result, or a new and useful method of producing an old thing or result.”¹⁹⁸

In so noting, the Court seems to have adopted the contrast made in that case between invention and discovery. This directly contradicts what the Court itself said about the distinction in an oft-cited passage in which it is said that it is “not decisive – it is not even helpful – to point out that beyond discovery of a scientific fact nothing has been added”¹⁹⁹ In support, their Honours marshalled the following quote from *Funk Bros* (discussed above):

It only confuses the issue, to introduce such terms as ‘the work of nature’ and the ‘laws of nature’. For these are vague and malleable terms infected with too much ambiguity and equivocation. Everything that happens may be deemed ‘the work of nature’, and any patentable composite exemplifies in its properties ‘the laws of nature’.²⁰⁰

Holding “laws of nature” to be a synonym for “discovery”, the Court went on to say:

The truth is that the distinction between discovery and invention is not precise enough to be other than misleading in this area of discussion. There may indeed be a discovery without invention – either because the discovery is some piece of abstract information without any suggestion of a practical application of it to a useful end, or because its application lies outside the realm of ‘manufacture’.²⁰¹

In support of this proposition, the Court clearly relied on criticisms similar to those made of Swinfen-Eady J by the appeal court in *Hickton's* case, arguing that “[t]he fallacy lies in dividing up the process that he puts forward as his invention. It is the whole process that must be considered”.²⁰²

¹⁹⁶ *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 262.

¹⁹⁷ *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 263.

¹⁹⁸ *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 263.

¹⁹⁹ *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 263.

²⁰⁰ *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 263-264.

²⁰¹ *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 264.

²⁰² *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 264.

Chapter 2: The History of Patentable Subject Matter

Yet their Honours discussion of *Lane Fox* clearly relies on a distinction between discovery and invention. This cannot have been an oversight on the part of the Court, since the passage quoted above basing the distinction on “knowledge and ingenuity” is taken directly from a paragraph of *Lane Fox* which begins “An invention is not the same thing as a discovery.”²⁰³ Thus subsequent cases which have relied on *NRDC* as authority for setting aside such a distinction entirely, have overstated the position.

Next their Honours then noted Lindley J’s purported requirement that a patentable process either have novelty in the mode of use as distinguished from novelty of purpose, or in the alternative that any “user, or modification, or appliance” involve appreciable merit. The Court however limited his Honour’s remarks to the context in which they were made, namely in dealing with “a thing which is known – not only the existence of which .. but the characteristics and properties [as well]”.²⁰⁴ As such there was

nothing in the judgment of Lindley LJ to justify a denial that, in respect of a process for achieving a useful result by the employment of a substance to produce effects which antecedently it was not understood to be capable of producing, the inventiveness which is essential for a valid grant of a patent may be found in the step which consists of suggesting the use of the thing for the new purpose, notwithstanding that there is no novelty or “appreciable merit” in any suggested mode of using the thing, or any modification of the thing or of an appliance necessary for using it for the new purpose.²⁰⁵

Scope of patentable subject matter

The Court then turned their attention to the central issue in the case – how to define the scope of patentable subject matter,²⁰⁶ or in other words, “whether the process that is claimed falls within the category of inventions to which, by definition, the application of the *Patents Act* is confined.”²⁰⁷ The Court set out the arguments of counsel as follows:

The Commissioner ... emphasizes the word “manufacture” and contends for an interpretation of it which, though not narrow, is restricted to vendible products and processes for their production, and excludes agricultural and horticultural products. ... The appellant, on the other hand, urges upon us a wider view: that there is a “manufacture” such as might properly have been the subject of letters patent and grant of privilege under s.6 of the *Statute of Monopolies* whenever a process produces, either immediately or ultimately, *a useful physical result in relation to a material or tangible entity*.²⁰⁸

²⁰³ *Lane Fox v Kensington and Knightsbridge Electric Lighting Company* (1892) 3 Ch. 424 per Lindley LJ at 428-429, discussed at page 50 above.

²⁰⁴ *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 263.

²⁰⁵ *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 263.

²⁰⁶ *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 268.

²⁰⁷ *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 268.

²⁰⁸ *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 268 (emphasis added).

The Court then set out in detail what they believed to be the proper approach to determining patentability. Because of its importance, the passage is set out in its entirety:

It is of the first importance to remember always that the *Patents Act* 1952-1955 (Cth), like its predecessor the *Patents Act* 1903 (Cth) and corresponding statutes of the United Kingdom (see the *Patents, Designs and Trade Marks Act* 1883, s.46; the *Patents Act* 1907, s. 93; and the *Patents Act* 1949, s.101) defines the word ‘invention’, not by direct explication and in the language of its own day, nor yet by carrying forward the usage of the period in which the *Statute of Monopolies* was passed, but by reference to the established ambit of s. 6 of that Statute. The inquiry which the definition [of patentable subject matter] demands is an inquiry into the scope of the permissible subject matter of letters patent and grants of privilege protected by the section. It is an inquiry not into the meaning of a word so much as into the breadth of the concept which the law has developed by its consideration of the text and purpose of the *Statute of Monopolies*. . . . The word “manufacture” finds a place in the present Act, not as a word intended to reduce a question of patentability to a question of verbal interpretation, but simply as the general title found in the *Statute of Monopolies* for the whole category under which all grants of patents which may be made in accordance with the developed principles of patent law are to be subsumed. It is therefore a mistake, and a mistake likely to lead to an incorrect conclusion, to treat the question whether a given process or product is within the definition as if that question could be restated in the form: ‘Is this a manner (or kind) of manufacture?’ It is a mistake which tends to limit one’s thinking by reference to the idea of making tangible goods by hand or machine, because ‘manufacture’ as a word of everyday speech generally conveys that idea. The right question is ‘Is this a proper subject of letters patent according to the principles which have been developed for the application of s. 6 of the *Statute of Monopolies*?’

It is a very different question. A perusal of the definitions and quotations appearing in the *Oxford English Dictionary* under ‘manufacture’ will show that the word has always admitted of applications beyond the limits which a strict observance of its etymology would suggest, and, as the present Chief Justice said in *Maeder v Busch*, a widening conception of the notion has been a characteristic of the growth of patent law.²⁰⁹

Perhaps aware of the fact that *Maeder v Busch* contained no direct authority for such a claim, their Honours looked for historical support for the notion of a widening conception of patent law. In support of this proposition their Honours turned to the judgment of Eyre CJ in *Boulton v Bull*, who the Court noted had said

that ‘the exposition of the statute as far as usage will expound it, has gone much beyond the letter’; and the width of the meaning may be gauged from the statement of the same learned judge that ‘manufacture’ extended ‘to any new results of

²⁰⁹ *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 269-270.

Chapter 2: The History of Patentable Subject Matter

principles carried into practice ... new processes in any art producing effects useful to the public’²¹⁰

In light of the uncertainty inherent in the four differing judgments in *Boulton v Bull*, their Honours went on to note that “[b]y 1842 it was finally settled that ‘manufacture’ was used in the *Statute of Monopolies* in the dual sense which comprehends both a process and a product: *Crane v Price*.”²¹¹

Their Honours later summed up the proper approach as follows:

The truth is that any attempt to state the ambit of s. 6 of the *Statute of Monopolies* by precisely defining ‘manufacture’ is bound to fail. The purpose of s. 6, it must be remembered, was to allow the use of the prerogative to encourage national development in a field which already, in 1623, was seen to be excitingly unpredictable. To attempt to place upon the idea the fetters of an exact verbal formula could never have been sound. It would be unsound to the point of folly to attempt to do so now, when science has made such advances that the concrete applications of the notion which were familiar in 1623 can be seen to provide only the more obvious, not to say the more primitive, illustrations of the broad sweep of the concept.²¹²

It was thus clearly intended to resist a reductionist approach to defining patentable subject matter. Instead, the Court were at great pains to keep the focus upon the historical development of patent law, and to direct the future course of patentable subject matter by reference to existing principles. It is therefore obviously wrong to treat the case as a break with the past. It is worth pointing out at this stage that the Court, despite supporting a broad and flexible approach to the determination of patentable subject matter, did not intend to remove all constraint. Their Honours specifically acknowledged the “[t]he need for qualification ... even if only in order to put aside, as they apparently must be put aside, processes for treating diseases of the human body...”²¹³

Having established the proper course of the inquiry, their Honours then turned their attention to Morton J’s “vendible product” test from *Re GEC*. Summaries of the case usually characterise the court’s treatment of this test as having thrown it out, or set it aside. No doubt this implication is to be drawn from the negative treatment the court gave to the notion of an “exact verbal formula”.²¹⁴ However, it would be more correct to characterise the Court’s treatment of it as a reconciliation. In fact, the “formula” which subsequent cases have referred to as summarising the correct approach to patent-eligibility in Australia is in fact directly derived from this test.

²¹⁰ *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 270 (footnotes omitted).

²¹¹ *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 270 (footnotes omitted).

²¹² *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 271.

²¹³ *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 270. Note that their Honours later suggested in *obiter* that such methods “may well lie outside the concept of invention because the whole subject is conceived as essentially non-economic”. *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 275. Specific limiting factors are considered in the section on vendible products below.

²¹⁴ *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 271.

It is valuable for its insistence that in patent law at the present day a process may be within the concept of ‘manufacture’ notwithstanding that it merely improves, restores or preserves some antecedently existing thing; but in so far as it may appear to restrict the concept by its use of the expression “vendible product”, it must be considered now as substantially qualified by the comments made upon it by Evershed J [in *Re Cementation*] and [Rantzen] ...²¹⁵

Their Honours then noted that in *Re Cementation* Evershed J had said

that the word ‘product’ was used in a sense which included ‘that which is produced by any action, operation or work; a production; the result’; so that it denoted the subject matter of each of the three forms of activity [namely production, improvement or restoration], and was not intended to limit the conception by reference to the common acceptance of ‘product’.²¹⁶

Rantzen’s case was taken to be relevant because of Evershed J’s interpretation of manufacture “not by rejecting Morton J’s mode of describing the ambit of the word, but by interpreting his expression ‘vendible product’ in a sense wide enough to include electrical energy, despite its non-material character, because of its analogy, in commercial respects, with material commodities. That this was sound is hardly to be doubted. In the varying applications of which the word ‘manufacture’ is capable analogy has always played a considerable part.”²¹⁷

The Court then sought to distinguish *Re Standard Oil Development Co.’s Application*²¹⁸ in which Lloyd-Jacob J had refused a patent for a selective herbicide on the basis that it could not be reconciled with Morton’s “rules”. Their Honours dismissed Lloyd-Jacob J’s reasoning as “hardly sufficient”.²¹⁹ Their Honours did not accept that “converting a weed-infested area to a weed-free area”²²⁰ was any different to “the process of converting a combustible subterranean formation into a non-combustible formation, or making a building fire-proof building”²²¹ and held:

Once it is conceded that land may be a “product” within the sense of Morton J’s “rule” as now understood, and that accordingly a process for improving it may be a “manufacture” in the relevant sense of the word ... a considerable step seems to have been taken towards establishing that an artificial process for suppressing unwanted forms of growth which impede the profitable use of land may be within the concept.

²¹⁵ *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 271-272

²¹⁶ *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 272.

²¹⁷ *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 272.

²¹⁸ (1951) 68 RPC 114.

²¹⁹ *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 274.

²²⁰ *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 274.

²²¹ *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 274.

Chapter 2: The History of Patentable Subject Matter

Their Honours then found, by reference to earlier cases, the notion that an improvement could be negative, in the sense of removing something from the “vendible product” in question.²²² In particular, the Court relied on *Elton and Leda Chemicals Ltd’s Application*²²³ as supporting the notion, consistent with the earlier decisions of Evershed J²²⁴ that the “vendible” part of the vendible product “test” should be taken as

laying the proper emphasis upon the trading or industrial character of the processes intended to be comprehended by the Acts – their ‘industrial or commercial or trading character’ as *Lloyd-Jacob J* himself described it in *Re Lenard’s Application*. The point is that a process, to fall within the limits of patentability which the context of the Statute of Monopolies has supplied, must be one that offers some advantage which is material, in the sense that the process belongs to a useful art as distinct from a fine art ... that it’s value to the country is in the field of economic endeavour.²²⁵

During the discussion of the development of the law in this Chapter, it has been shown how a physicality requirement has again and again formed part of the test for inherent patentability both explicitly and implicitly in various cases. However the Court in *NRDC* had expressed an equivocal view as to such a requirement:

But a question which appears still to await a final decision is whether it is enough that a process produces a useful result or whether it is necessary that some physical thing is either brought into existence or so affected as the better to serve man’s purposes. In some of the cases it is suggested that the process must issue in some ‘vendible matter’ or a ‘vendible product’. The former expression was used by Heath J in *Boulton v Bull* in the course of maintaining the opinion, which must now be considered heretical, that there could not be a patent for a method; but no such expression appears in the powerful judgment in which Eyre CJ maintained the opposite view and reached the conclusion in the particular case which was ultimately upheld in *Hornblower v Boulton*. Abbott CJ in *R v Wheeler* having spoken of a ‘thing made, which is useful for its own sake, and vendible as such’ went on to show that he did not find in such expressions such as those any absolute test. He said (the italics are ours):

‘Something in the nature of a corporeal and substantial nature, something that can be made by man from the matters subjected to his art and skill, *or at the least some new mode of employing practically his art and skill*, is requisite to satisfy this word’.

It is of course not possible to treat such a statement as conclusive of the question.²²⁶

²²² *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 275-276. In support their Honours cited *Hall v Jarvis* (1822) 1 Web PC 100 (removing superfluous and loose fibres); *Re Alsop’s Patent* (1907) 24 RPC 733 (chemically cleaning dirty linen)

²²³ [1957] RPC 267 (method of dispersing fog).

²²⁴ *Re Application by Rantzen* (1946) 64 RPC 63 at 66.

²²⁵ *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 275.

²²⁶ *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 270.

Later in the judgment however, the Court in considering the second limb of the vendible product test, referred to *Elton and Leda Chemicals*,²²⁷ before stating:

the tenor of the passage seems to be that what is meant by a ‘product’ in relation to a process is only something in which the new and useful effect may be observed. Sufficient authority has been cited to show that the ‘something’ need not be a ‘thing’ in the sense of an article; it may be *any physical phenomenon in which the effect*, be it creation or merely alteration, *may be observed*...²²⁸

Then their Honours completed the reconciliation of vendible product with the correct approach as follows:

It is, we think, only by understanding the word ‘product’ as covering ‘every end produced’, and treating the word ‘vendible’ as pointing only to the requirement of utility in practical affairs, that the language of Morton J’s “rule” may be accepted as wide enough to convey the broad idea which the long line of decisions on the subject has shown to be comprehended by the Statute.²²⁹

Application to the present case

Using the approach to locating inventiveness they derived from *Lane Fox*, the Court later characterised the invention as follows:

by an application of scientific ingenuity, combining knowledge, thought and experimentation, not only in relation to the chemicals but in relation also to the enzyme systems of certain weeds and plants, the applicant has evolved a new and useful method of destroying weeds without harming useful vegetation amongst which they are growing. It is irrelevant, even if true, that once the discovery was made that the chemicals produce a lethal reaction when applied to the weeds and produce no such reaction when applied to the crops there was no more ingenuity required in order to show how the process might be performed. The point that matters is that a weed killing process involved a step plainly inventive.²³⁰

For the Court, the inventiveness in the present application lay in the fact that

the process differs from the previously known processes of its kind in this, that it employs substances the suggestion of which for the purpose in hand was new, was not obvious, and was to be arrived at only by an exercise of scientific ingenuity,

²²⁷ *Elton and Leda Chemicals Ltd's Application* [1957] RPC 267 at 268-269, cited in *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 276.

²²⁸ *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 276 (emphasis added).

²²⁹ *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 276.

²³⁰ *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 265.

Chapter 2: The History of Patentable Subject Matter

based upon knowledge and applied in experimental research. The fact that the substances themselves were already known to man affords no valid reason for denying that the suggestion was inventive.²³¹

Their Honours then applied the revised vendible product test to the present claimed invention:

It is a 'product' because it consists in an artificially created state of affairs, discernible by observing over a period the growth of weeds and crops respectively on sown land in which the method has been put into practice. And the significance of the product is economic; for it provides a remarkable advantage, indeed to the lay mind a sensational advantage. ... The method cannot be classed as a variant of ancient procedures. It is additional to the cultivation. It achieves a separate result, and the result possesses its own economic utility consisting in an important improvement in the conditions in which the crop is to grow. ...²³²

It is from this passage that the so-called "NRDC test" is distilled, namely the twin requirements of an "artificially created state of affairs" and "economic utility". However, rather than giving these phrases any particular literary significance or their meaning being "a question of verbal interpretation",²³³ it is submitted that these terms are just convenient ways of referring to particular aspects of the "principles which have been developed for the application of section 6 of the Statute of Monopolies".²³⁴

One problem which a reductionist account of NRDC creates is that it glosses over the many ways in which the underlying principles are expressed. "Vendibility" is variously expressed by the Court as:

- Being a useful art as opposed to a fine art;
- Having a trading or industrial character;
- Being of value in the field of economic endeavour; and
- Having economic utility.

Similarly, the "product" requirement is expressed as:

- Every end produced;
- A thing made;
- Physical phenomenon in which a useful effect may be observed;
- Having a corporeal or substantial nature

²³¹ *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 268.

²³² *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 277.

²³³ *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 269.

²³⁴ *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 269.

Each of these different expressions capture different nuances of what it is for something to fall within the scope of patent law. The differences in the way these requirements are expressed may be more or less useful in a particular case. Also, each form of words captures something of the case from which it is described, and is in itself shorthand for the fuller expressions of the judges who have developed patent law since before 1623. As such, any claim that one particular expression should be given primacy over the others is to go against the Court's caution about the folly of looking for an "exact verbal formula", and the direction of the Court that inherent patentability should be understood by reference to "the principles which have been developed for the application of section 6 of the *Statute of Monopolies*". That is, they should be afforded no more primacy than "vendible product" which could be similarly understood.

Another point flows from this submission. There is a tendency to rely on *NRDC* as an excuse to treat the inherent patentability issue in a permissive fashion. This is something which is easily read into the language used in the case, concerned as it was with overriding what was considered to be an unduly narrow approach centred around a strict interpretation of Morton J's rules. However, the Court emphasised both explicitly, and by reference to the elucidation of the requirements set out above, that qualifications are necessary. In other words, the flexibility which the case allows should not be reinterpreted as a strong bias towards patentability. The very flexibility which is claimed as central to the success of the formula is potentially a weak point, in that the component terms of the formula can be interpreted in such a way that they could extend to any claimed invention. The effect of such an approach would be to disrupt the *quid pro quo* of the patent regime – disclosure in the public interest in exchange for protection of the invention, as first espoused in *Liardet v Johnson*. The public interest is not served by awarding monopolies to things which are not in the public interest. This is a point which will be further developed later in this thesis.

7 Conclusion

This chapter has set out the development of the approach to patentability in Australia, the United Kingdom and the United States, up to the time of *NRDC*. From the development of the case law on patentable subject matter to *NRDC*, a number of themes emerge. From the early patent custom, and circumstances surrounding the passage of the *Statute of Monopolies*, the importance of the patent monopoly as a means of establishing industry can be gleaned – a purpose patent law still attempts to achieve to this day. Secondly, it can be seen how the law has had to adapt to meet the changing nature of invention over the last 400 years, beginning with entire trades and devices in the early years, to cover more and more abstract subject matter. Despite this, the principles to be applied are broadly consistent with those espoused (albeit in opposition to each other) in *Boulton v Bull*: that principles or other abstract notions are not patentable; and that process patents can be problematic, no doubt due to their abstract nature.

As the technological developments have tended towards more abstract subject matter, the simplistic distinctions between the patentable and the non-patentable have come under closer scrutiny. A bare denial of the patentability of principles was complicated by the recognition that

Chapter 2: The History of Patentable Subject Matter

an application thereof to a specific purpose is required. This in turn lead to the development of further ways of distinguishing the border line, through such limiting factors as requiring a physical manifestation; allowing patents only on the means rather than ends achieved; and a trading, industrial or commercial character. These requirements were expressed in a range of different fashions, depending on the context in which they arose, and each re-framing of the nature of patentable subject matter adds to the understanding of this difficult area not by providing a new and improved formula for defining the scope of patentability, but by capturing nuances which may be of more help in some cases than in others. The multifaceted nature of the patentable subject matter inquiry was captured by the judgment in *NRDC*, which added its own "reformulation" which although attractive as a fresh starting point, is necessarily informed and fleshed out by the previous development of the law upon which it depends.

The next chapter will continue the development of the law of patentable subject matter, as it has been applied specifically to computer software. It will be shown how an expansive approach to patentable subject matter followed in the wake of *NRDC*, both in Australian law, but also in the US and UK. Tracking the patentability of software to the present time also evidences how the flexibility of the subject matter inquiry has been stretched to breaking point, and how the most recent cases suggest a course correction towards a more narrow interpretation of what should be patentable and what is not.

How Software Became Patentable

*“Analogies can be suggestive, even illuminating. But when lawyers and judges reason analogically from old cases to new, it usually means that they are limiting their analysis of the present case to what can be found in prior cases; and ... the prior cases often constitute an impoverished repository of fact and policy for the decision of the present one.”*¹

1 Introduction

Having introduced the key features of computer software in Chapter 1, and outlined the historical development of the patentable subject matter inquiry in Chapter 2, it is now possible to draw these two topics together. This Chapter begins by briefly setting out the early history of modern computing, and the emergence of software as a separate phenomenon to the hardware upon which it runs. The Chapter then charts the course of software from non-patentable idea (or mental process) to patentable process in all 3 jurisdictions, despite a clear exclusion in the UK. The Chapter concludes by analysing the current state of the patentable subject matter inquiry, both as it relates to software, and more broadly for other abstract subject matter. Such an understanding is essential before moving into the subject matter of the next chapter, the problems with software patents.

2 The emergence of the modern computer

As noted in the earlier chapters, the history of computing is largely one of computer hardware. Computer hardware owes its origins to the mechanical calculator, the first of these, Schickard’s Calculating Clock, which could add and subtract six-digit numbers,² came into existence at around the same time as the *Statute of Monopolies*. Such a device would clearly have come

¹ Richard A Posner, *Overcoming Law* (Harvard University Press, 1995) at 519.

² Encyclopaedia Britannica, “Calculating Clock” *Encyclopaedia Britannica Online*, <<http://www.britannica.com/EBchecked/topic-art/725523/19230/The-Calculating-Clock-A-reproduction-of-Wilhelm-Schickards-Calculating-Clock>> (19 November 2009).

Chapter 3: How Software Became Patentable

within the notion of manner of manufacture described above, as it was a physical machine, just like a steam engine.

From the Calculating Clock onwards, various devices had been designed and constructed which allowed the solution of mathematical problems. However, these were designed and built in an *ad hoc* manner, and in the absence of any theoretical framework. In the 1930s, this began to change. In 1931, the publication of Kurt Gödel's Incompleteness Theorem³ led to the crystallisation of the theoretical basis of computer science in 1936 by Alonzo Church and Alan Turing.⁴ At around the same time, Claude Elwood Shannon formulated the concept upon which all modern computers are built, namely that of using electrical relays and switches to solve logic problems.⁵

The first computers began to emerge in the period 1939-1941.⁶ The focus was on building machines that could assist people to complete calculations faster, and in particular the need to perform large calculations for wartime purposes, such as ballistics, cryptography and troop deployments.⁷

However, "their numerical processing capabilities were soon applied to more general logical manipulations".⁸ One limitation of these early machines was the way in which the various components making up the machine had to be physically wired together, meaning that "re-programming" was a time-consuming physical task. In 1945, in response to such limitations, von Neumann came up with his influential "stored program" architecture, giving birth to the hardware-software divide. Von Neumann suggested that hardware should be kept simple and controlled by complex software, in order that the computer could be reprogrammed more efficiently.⁹ The first stored-program computer was built around 1948.¹⁰ Australia's first computer, the CSIR Mk.1, became operational just over 1 year later.¹¹ The first commercial computers were sold in around 1951.¹² The first non-hardware company offering commercial program-

³ This topic is explored in detail in Chapter 8.

⁴ These key developments are studied in great detail in Chapter 8.

⁵ See Claude E Shannon, *A symbolic analysis of relay and switching circuits* (Massachusetts Institute of Technology, Dept. of Electrical Engineering, 1940).

⁶ There is no single "first computer", with many variations on a theme developed in parallel at this time. Notable early examples were the Zuse Z3, the Colossus, the Atanasoff-Berry Computer and ENIAC. A useful overview can be found at the Computer History Museum, "Timeline of Computing" <<http://www.computerhistory.org/timeline/?category=cmptr>> (9 March 2010). For more detail, see "History of Computing Hardware", *Wikipedia* <http://en.wikipedia.org/wiki/History_of_computing_hardware> (9 March 2010).

⁷ Roger Clarke, "Origins and Nature of the Internet in Australia" <<http://www.anu.edu.au/people/Roger.Clarke/II/OzI04.html>> (9 Mar 2010); Kevin Spaulding, "Brief History of Computers: A Look Back at Computing" <<http://knol.google.com/k/brief-history-of-computers>> (9 March 2010).

⁸ See Roger Clarke, "Origins and Nature of the Internet in Australia" <<http://www.anu.edu.au/people/Roger.Clarke/II/OzI04.html>> (9 Mar 2010); (refs?)

⁹ John von Neumann, *First Draft of a Report on the EDVAC* (Moore School of Electrical Engineering, University of Pennsylvania, 30th June 1945) <<http://qss.stanford.edu/~godfrey/vonNeumann/vnedvac.pdf>> (9 March 2010).

¹⁰ This was the Manchester Small-Scale Experimental Machine (SSEM) also known as "Baby". See Brian Napper, "The Manchester Small Scale Experimental Machine – 'The Baby'" <<http://www.computer50.org/mark1/new.baby.html>> (14 September 2011); *Wikipedia*, "Stored Program" <http://en.wikipedia.org/wiki/Stored_program> (7 September 2009).

¹¹ See Roger Clarke, "A retrospective of the information systems discipline in Australia" in Guy G. Gable (ed), *The Information Systems Academic Discipline in Australia* (1988) 75 at 85; *Wikipedia*, "CSIRAC" <<http://en.wikipedia.org/wiki/CSIRAC>> (7 September 2008).

¹² These were the UNIVAC, and the IBM 701. See "History of Digital Computers: Since 1950" in Anthony Ralston, Edwin D Reilly and David Hemmendinger (eds), *Encyclopedia of Computer Science* (4th ed, Wiley, 2003).

The emergence of the modern computer

ming services in the US was founded four years later in 1955,¹³ although the first computer-program product was not sold until the mid 1960s.¹⁴

Whilst a complete history of modern computing is more thoroughly addressed elsewhere,¹⁵ brief mention should be made of the series of milestones which contributed to the modern ubiquity of computers, in all their forms. These are:

- the invention of the transistor in 1948,¹⁶ which greatly reduced the size, expense and reliability of computers. IBM's transistor-based 7000 series, introduced in 1956, saw it gain 81% of market share by 1961.¹⁷
- development of the integrated circuit (IC) in 1958, allowing for multiple components to be combined on one chip.¹⁸ The advantage of the IC was that it removed the need for hand-wiring between components. Integrated circuits became commercially available in 1961,¹⁹ and were in use in computers by Texas Instruments in 1962.²⁰ Although the first IC contained only 5 components, refinement of the technology led to the introduction of the microprocessor by Intel in 1971.²¹ A microprocessor is an entire computer fabricated on one chip.
- the rise of the personal microcomputer market in 1975.²² The MITS Altair, a mail-order DIY computer, although technically not the first small-scale computer,²³ captured public attention and defined the market for such devices.²⁴

¹³ The Computer Usage Corporation was founded in 1955 by two former IBM employees: "The Software Industry in the 1950s" <http://www.softwarehistory.org/history/d_50s.html> (9 March 2010).

¹⁴ Martin Campbell-Kelly, "Not All Bad: An Historical Perspective on Software Patents" (2005) 11 *Michigan Telecommunications and Technology Law Review* 191 at 191.

¹⁵ For a brief illustrated history, see Computer History Museum, "Timeline of Computing" <<http://www.computerhistory.org/timeline/?category=cmptr>> (9 March 2010); John Kopplin, "An Illustrated History of Computers" *ComputerScienceLab.com*, 2002

<<http://www.computersciencelab.com/ComputerHistory/History.htm>> (10 March 2010); . For a more detailed account, see Martin Campbell-Kelly and William Aspray, *Computer: A History of the Information Machine* (1998).

¹⁶ Mary Bellis, "The History of the Transistor – John Bardeen, Walter Brattain and William Shockley" *About.com* <<http://inventors.about.com/library/weekly/aa061698.htm>> (10 March 2010).

¹⁷ Computer History Museum, "Timeline of Computing" <<http://www.computerhistory.org/timeline/?category=cmptr>> (9 March 2010).

¹⁸ Mary Bellis, "The History of the Integrated Circuit aka Microchip" *About.com* <http://inventors.about.com/od/istartinventions/a/intergrated_circuit.htm> (10 March 2010).

¹⁹ Mary Bellis, "The History of the Integrated Circuit aka Microchip" *About.com* <http://inventors.about.com/od/istartinventions/a/intergrated_circuit.htm> (10 March 2010).

²⁰ Mary Bellis, "The History of the Integrated Circuit aka Microchip" *About.com* <http://inventors.about.com/od/istartinventions/a/intergrated_circuit.htm> (10 March 2010).

²¹ Mary Bellis, "Inventors of the Modern Computer: Intel 4004 – The World's First Single Chip Microprocessor" *About.com* <<http://inventors.about.com/od/mstartinventions/a/microprocessor.htm>> (10 March 2010).

²² See for example "Altair" <<http://www.cis.usouthal.edu/faculty/daigle/project1/1975malt.htm>> (10 March 2010).

²³ The plan for the first small-scale computer, named Simon, was first conceived of in 1949: see Edmund Berkeley, *Giant Brains, or Machines that Think* (1949). Plans for Simon were published in *Radio Electronics* in 1950 and 1951. See "Pop Quiz: What was the first personal computer?" *Blinkenlights Archaeological Institute* <<http://www.blinkenlights.com/pc.shtml>> (10 March 2010).

²⁴ Although its makers only envisaged selling a few hundred to hobbyists, thousands of 8800s were ordered in the first month: "March of the outdated machines" *New Scientist*, 21 September 2009 <<http://www.newscientist.com/gallery/dn17805-computer-museums-of-the-world/4>> (10 March 2010). For more information about the product, see Wikipedia, "Altair 8800" <http://en.wikipedia.org/wiki/Altair_8800> (10 March 2010).

Chapter 3: How Software Became Patentable

- the introduction of the IBM PC running Microsoft DOS in 1981. The widespread adoption of this single architecture arguably led to today's mass market for software.²⁵

As computers became more powerful, simpler calculations started to be assembled into more complicated sets of calculations that started to have a value of their own. This was greatly aided by the invention of the compiler in 1952, the operating system in 1956 and the first high-level programming language, FORTRAN, in 1957.²⁶ The term software was coined in 1958.²⁷ By 1965, the software industry had grown significantly. By that time "there were an estimated 45 major software contractors in the U.S, some employing more than a hundred programmers and with annual revenues as much as \$100 million."²⁸

Campbell-Kelly describes the emergence of an independent software industry over the same period as follows:

In the mid-1950s, about a dozen firms entered the programming services industry, writing programs to order for clients. Here again, software protection was not a significant issue because programs written under contract for an organization were so particular that they would have had little value to another organization. Moreover, in an era when computers cost several hundred thousand dollars a year to rent, the high programming cost was buried under the overall cost of computer ownership.

The situation changed in the mid 1960s with the arrival of the IBM System/360 computer. The System/360 was extremely successful, creating for the first time an industry standard platform. At the same time, the computer population had begun to explode—from 5,500 worldwide in 1960 to 29,600 by 1965, an annual growth rate of 30 percent. Falling hardware prices had created a new sector of the computer market of corporations paying annual rentals of as little as fifty thousand dollars. For these new owners, custom-written programs were not economically justifiable. Fortunately, the new generation of computers had much greater speeds and larger memories than previous models, allowing the inefficiencies of generalized software "products" to be tolerated rather than requiring development of more efficient, custom-written programs. For software firms, the large customer base of System/360 users made writing such generalized programs a viable business proposition.²⁹

²⁵ See Wikipedia, "Influence of the IBM-PC on the personal computer market" <http://en.wikipedia.org/wiki/Influence_of_the_IBM-PC_on_the_personal_computer_market> (30 September 2010).

²⁶ Computer History Museum, "Timeline of Computing" <<http://www.computerhistory.org/timeline/?category=cmptr>> (9 March 2010)

²⁷ Ivars Peterson "Software's Origin" *Ivar Peterson's MathTrek Archive*, 31 July 2000 <http://www.maa.org/mathland/mathtrek_7_31_00.html> (10 March 2010); David Leonhardt, "John Tukey, 85, Statistician; Coined the Word 'Software'", *New York Times*, 28 July 2000, at A19

²⁸ "The Software Industry in the 1960s" <http://www.softwarehistory.org/history/d_60s.html> (9 March 2010).

²⁹ Martin Campbell-Kelly, "Not All Bad: An Historical Perspective on Software Patents" (2005) 11 *Michigan Telecommunications and Technology Law Review* 191, at 211.

3 How software became patentable subject matter

As the software industry developed, so too was patent law developing in parallel. As discussed in Chapter 2, the development of the inherent patentability inquiry has been subject to ongoing reformulation and adaptation since it first came into being. The most recent adaptation in Australia was the re-framing of the inquiry in *NRDC*, which many saw as a break with the past, but which it has been shown only served to reinforce its historical foundations.

Unfortunately, judicial consideration of the proper application of the *NRDC* approach to software-related inventions was a long time coming. The issue was first considered by a single judge of the Federal Court in 1991.³⁰ The Full Federal Court did not weigh in until 1996.³¹ The High Court has never turned its attention to the problem. Thus to properly understand how software became patentable, it is necessary to look to case law abroad to relevant developments in the US and UK over this period.

The global dominance in computer hardware and software development of the United States might suggest that after 40 years of litigation, and the associated development of jurisprudence, the issue might be resolved. Until recently, it seemed as though the issue had been resolved in favour of the unfettered patenting of software.³² As a counter point we have the United Kingdom, whose initially favourable view of the patentability of software was turned on its head by the passage of the European Patent Convention,³³ which expressly excluded computer programs “as such” from patentable subject matter.³⁴ As such it provides a counter point to the US perspective, although the subsequent watering-down of the exclusion, especially at the European Patent Office, highlights the difficulty in defining the scope of patentable subject matter, either in positive or negative terms.

The United States

Before turning to the development of the case law itself, it is apposite to say something about the way in which patent law is structured in the US. Patent applications once lodged are considered by an examiner at the United States Patent and Trademark Office (“USPTO”). Decisions of patent examiners are subject to review within the USPTO by a quasi-judicial body, the Board of Patent Appeals and Interferences (“BPAI”). As patent law is a federal law, the US federal court system has jurisdiction over patent matters. Thus decisions of the examiners and the BPAI lie to either a United States District Court, or Court of Appeals. From 1929 until 1982, appeals from the Patent Office were heard in the Court of Customs and Patent Appeals (“CCPA”). On 1 October 1982, a new Court of Appeals for the Federal Circuit (“CAFC”) was formed in order

³⁰ *IBM v Commissioner of Patents* (1991) 33 FCR 218 discussed below at 121.

³¹ *CCOM Pty Ltd v Jiejing Pty Ltd* (1994) 51 FCR 260, discussed below at 123.

³² Those recent developments are discussed in Section 4 below.

³³ *Convention on the Grant of European Patents*, opened for signature 5 October 1973, 13 ILM 268 (entered into force 7 October 1973).

³⁴ The relevant terms of the European Patent Convention, and the administrative changes it effected are discussed in Section 3 below.

Chapter 3: How Software Became Patentable

to bring greater uniformity to patent law.³⁵ Appeals from district courts lie to the CAFC, and from the CAFC to the Supreme Court of the United States of America (“SCOTUS”), although leave must be granted before such an appeal is heard.

Mental steps doctrine

As the age of computing began apace, the automation of processes previously carried out by humans began to raise specific questions as to the relationship between non-patentable ideas and patentable processes.³⁶ As early as *O’Reilly v Morse*, patents had been awarded for processes requiring human mental activity in their performance.³⁷ At the other end of the scale however, lay cases such as *Ex parte Meinhardt*³⁸ wherein the Commissioner of Patents refused an application for a system for spacing freehand letters practiced entirely by a human. The Commissioner held that the only two classes of patentable process were those involving chemical or other action, and those of a mechanical nature, a hat-tip to the *Cochrane v Deener* transformation requirement.³⁹ Similarly, the Court in *Greenewalt v Stanley Co of America*⁴⁰ held

We do not find authority in the law for the issuance of a patent for results dependent upon such intangible, illusory and non-material things as emotional or aesthetic reactions.⁴¹

The so-called “mental steps doctrine” arose out of the case of *Halliburton v Walker*.⁴² In that case the Court ruled ineligible a claimed method of determining the location of obstruction in an oil well. The Court emphasised the dependence on descriptive words in the claims, such as “‘determining,’ ‘registering,’ ‘counting,’ ‘observing,’ ‘measuring,’ ‘comparing,’ ‘recording,’ ‘computing,’”⁴³ holding that such “mental steps, even if novel, are not patentable”.⁴⁴

³⁵ Prior to the formation of the CAFC, appeals from decisions of the patent office went to the CCPA in Washington, but the CCPA did not have jurisdiction over infringement actions, which were heard in the local district court, and appeals from the district court went to the court of appeals in the district where the trial was held, whose rulings were not always consistent with each other. The new CAFC received the jurisdiction and judges of the CCPA, and in its first decision declared that it would only follow the precedents of the CCPA, and not those of the other circuit courts. See Gregory Stobbs, *Software Patents* (2nd ed, 2000) at 118-119; .

³⁶ That such a close relationship exists is borne out by the usage of the term “computer”, which was historically used to refer to a person who makes calculations or computations, and specifically someone employed for such a purpose. It was not until much later that the term was used as a descriptor of a computing machine. See “Computer”, *Oxford English Dictionary*, (2nd edition, 2009).

³⁷ See Norman D McClaskey, “The Mental Process Doctrine: Its Origin, Legal Basis, and Scope” (1969) 55 *Iowa Law Review* 1148 at 1151-1152.

³⁸ [1907] Dec. Com. Pat 237. Discussed in Norman D McClaskey, “The Mental Process Doctrine: Its Origin, Legal Basis, and Scope” (1969) 55 *Iowa Law Review* 1148 at 1162-1165.

³⁹ Discussed in Chapter 2. The machine-or-transformation test also reappeared with gusto in recent US case law, discussed below. See also Norman D McClaskey, “The Mental Process Doctrine: Its Origin, Legal Basis, and Scope” (1969) 55 *Iowa Law Review* 1148 at 1163-1164.

⁴⁰ 12 USPQ 122 (1931) .

⁴¹ *Greenewalt v Stanley Co of America* 12 USPQ 122 (1931) at 123. McClaskey suggests that the Court was concerned with unpredictable results of method relying on an emotional reaction for its performance as too vague and indefinite: Norman D McClaskey, “The Mental Process Doctrine: Its Origin, Legal Basis, and Scope” (1969) 55 *Iowa Law Review* 1148 at 1167.

⁴² *Halliburton Oil Well Cementing Co v Walker* 146 F.2d 817 (1944).

⁴³ *Halliburton Oil Well Cementing Co v Walker* 146 F.2d 817 (1944) at 821.

⁴⁴ *Halliburton Oil Well Cementing Co v Walker* 146 F.2d 817 (1944) at 821.

How software became patentable subject matter

The high water mark of the doctrine came in *Abrams*.⁴⁵ In rejecting an application for a method of prospecting for petroliferous deposits, the Court of Customs and Patent Appeals (CCPA), despite after noting various cases in support for the doctrine, held that

[c]itation of authority in support of the principle that claims to mental concepts which constitute the very substance of an alleged invention are not patentable is unnecessary. It is self-evident that thought is not patentable.⁴⁶

Abrams also became (in)famous for the so-called “rules” submitted to the Court by counsel for the Appellant. It was suggested:

1. If all the steps of a method claim are purely mental in character, the subject matter thereof is not patentable within the meaning of the patent statutes.
2. If a method claim embodies both positive and physical steps as well as so-called mental steps, yet the alleged novelty or advance over the art resides in one or more of the so-called mental steps, then the claim is considered non-patentable for the same reason that it would be if all the steps were purely mental in character.
3. If a method claim embodies both positive and physical steps as well as so-called mental steps, yet the novelty or advance over the art resides in one or more of the positive and physical steps and the so-called mental step or steps are incidental parts of the process which are essential to define, qualify or limit its scope, then the claim is patentable and not subject to the objection contained in 1 and 2 above.⁴⁷

In *dicta* the Court gave these rules notional support, indicating that “[f]rom such examination of the decisions as we have been able to make, the suggested rules appear to accord with them, but it is unnecessary for us arbitrarily to go beyond the requirements of the instant case.”⁴⁸ Despite this limited support, these “rules” were given an elevated role in the application of the doctrine.⁴⁹ *Abrams* was followed in *Yuan*,⁵⁰ with the CCPA holding in that case that a process requiring a human to solve mathematical equations was not patentable subject matter.⁵¹

Early US Patent Office practice in relation to computer programs did not focus on subject matter issues.⁵² By the 1960s, this practice started to come under pressure, as a number of cases had

⁴⁵ *In re Abrams* 89 USPQ (BNA) 266 (1951).

⁴⁶ *In re Abrams* (1951) 89 USPQ (BNA) 266 at 269.

⁴⁷ *In re Abrams* (1951) 89 USPQ (BNA) 266 at 267-8.

⁴⁸ *In re Abrams* (1951) 89 USPQ (BNA) 266 at 268.

⁴⁹ See *In re Prater* (1969) 415 F.2d 1378 at 1386: “[M]uch confusion in subsequent interpretation of the *Abrams* decision has been caused by people misreading the decision as conferring judicial sanction upon the ‘rules’ formulated and proposed by *Abrams*’ attorney.”

⁵⁰ *In re Yuan* (1951) 89 USPQ 324 at 329.

⁵¹ Cf *Ex parte Egan* (1960) 129 USPQ 23 (Board of Appeal) where a method of using a set of charts to allow an unskilled operator to arrive at a set of values normally required calculated by solving a mathematical problem was held to be patentable on the basis that it was “a process of using a machine or manufacture .. in a new way .. and yields advantageous results” at 26-27. Such a use was held to be “analogous to a method of operating a computer” at 26.

⁵² Pamela Samuelson, “Benson Revisited: The Case Against Patent Protection for Algorithms and Other Computer Program-Related Inventions,” (1990) 39 *Emory Law Journal* 1025 at 1039, citing in support *Ex parte King* (1964) 146 USPQ (BNA) 590. Cf Robert O Nimtz, “The Patentability of Computer Programs” (1970) 1 *Rutgers Journal of*

Chapter 3: How Software Became Patentable

“implied in passing that computer programs fell within statutory subject matter”.⁵³ To resolve the conflict, in 1966 a Presidential Commission in the United States held hearings and considered the patentability of software. The Commission was made up of “distinguished scientists, academics, and representatives of leading computer and high technology firms (as well as the Commissioner of Patents)”.⁵⁴ The Report recommended that Congress remove software from the ambit of the patent system.⁵⁵ The Commission noted that

[u]ncertainty now exists as to whether the statute permits a valid patent to be granted on programs. Direct attempts to patent programs have been rejected on the ground of nonstatutory subject matter. Indirect attempts to obtain patents and avoid the rejection, by drafting claims as a process, or a machine or components thereof programmed in a given manner, rather than as a program itself, have confused the issue further and should not be permitted.

The Patent Office now cannot examine applications for programs because of a lack of a classification technique and the requisite search files. Even if these were available, reliable searches would not be feasible or economic because of the tremendous volume of prior art being generated. Without this search, the patenting of programs would be tantamount to mere registration and the presumption of validity would be all but nonexistent.

It is noted that the creation of programs has undergone substantial and satisfactory growth in the absence of patent protection and that copyright protection for programs is presently available.⁵⁶

A bill giving effect to these recommendations was put before Congress, although it never became law.⁵⁷ Subsequently, the Patent Office issued a guideline expressly stating that computer programs, whether claimed as apparatuses or as methods,⁵⁸ were not suitable subject matter *per se*.⁵⁹ As the qualification suggests, the guidelines indicated that it may still be possible to patent a computer-program related invention in some circumstances:

Computers and Law 38 at 38-39, who relies the testimony of the Commissioner of Patents to the Presidential Commission that the Patent Office “had never knowingly issued a patent directed to a ‘computer program per se’”: *ibid* at 39, footnote 6.

⁵³ Harold L Davis, “Computer Programs and Subject Matter Patentability” (1977) 6 *Rutgers Journal of Computers and Law* 1, at 9, n44. Davis cites the following cases in support: *Ex parte Egan* 128 USPQ 23 (BNA, 1960); *Ex parte King* 146 USPQ 590 (BNA, 1964); *In re Naquin* 398 F.2d 863 (CCPA, 1968).

⁵⁴ Pamela Samuelson, “Benson Revisited: The Case Against Patent Protection for Algorithms and Other Computer Program-Related Inventions,” (1990) 39 *Emory Law Journal* 1025 at 1038.

⁵⁵ Presidential Commission on the Patent System, *To Promote the Progress of Useful Arts in an Age of Exploding Technology* (1966).

⁵⁶ Cited with approval in *Gottschalk v Benson* 409 US 63 (1973) at 72.

⁵⁷ The relevant section read “A plan of action or set of operating instructions, in whatever form presented, to cause a controllable data processor or computer to perform selected operations shall not be patentable.” Cited in Robert O Nimtz, “The Patentability of Computer Programs” (1970) 1 *Rutgers Journal of Computers and Law* 38 at 39.

⁵⁸ Pamela Samuelson, “Benson Revisited: The Case Against Patent Protection for Algorithms and Other Computer Program-Related Inventions,” (1990) 39 *Emory Law Journal* 1025 at 1040.

⁵⁹ “The basic principle ... is that computer programming per se ... shall not be patentable.”: (1966) 855 *Official Gazette of the Patent Office* 829: cited in Harold L Davis, “Computer Programs and Subject Matter Patentability” (1977) 6 *Rutgers Journal of Computers and Law* 1, at 9, n47. See also Robert O. Nimtz, “The Patentability of Computer Programs” (1970) 1 *Rutgers Journal of Computers and Law* 38 at 39.

How software became patentable subject matter

[a] computer programming process which produces no more than a numerical, statistical or other informational result is not directed to patentable subject matter. Such a process may, however, form a part of a patentable invention if it is *combined in an unobvious manner with physical steps* of the character [such as] in the knitting of a pattern or the shaping of metal.⁶⁰

The rejections of early software patents by the PTO lead to their consideration on appeal by the CCPA. The CCPA took a different view,⁶¹ turning its back on the doctrine it had developed in the earlier case and reversing the Patent Office position, and opening the door to the patenting of computer software.⁶² The key battleground between the Patent Office and CCPA was the extent to which computer programs were excluded from patentable subject matter as mental processes.⁶³ The Patent Office maintained that if a claimed process could be performed by a person with a pen or paper, it was not patent-eligible. The CCPA took a divergent view, although the exact basis for that view changed over the course of a series of cases.⁶⁴

In the case of *Prater*,⁶⁵ the CCPA analysed the development of the mental steps doctrine. “The CCPA asserted that its earlier decision, [Abrams] had been ‘misread’, that the other ‘mental process’ cases were ‘unsupported by precedent’ and could have been decided on other grounds, and that the Supreme Court’s decision in *Cochrane v Deener* ... had been ‘misconstrued’.”⁶⁶ The Court finally concluded that:

patent protection for a process disclosed as being a sequence or combination of steps capable of performance without human intervention and directed to an industrial technology – a ‘useful art’ within the intendment of the Constitution – is not precluded by the mere fact that the process could alternatively be carried out by mental steps.⁶⁷

An application for rehearing⁶⁸ sought by the Patent Office on the basis that the Constitutional issues had not been completely dealt with. In particular, the issue of mental steps and the First Amendment was raised,⁶⁹ with the Patent Office arguing that the subject patent would “confer

⁶⁰ (1968) 33 Fed. Reg. at 15,610 cited in Pamela Samuelson, “Benson Revisited: The Case Against Patent Protection for Algorithms and Other Computer Program-Related Inventions,” (1990) 39 *Emory Law Journal* 1025 at 1040, footnote 45.

⁶¹ The reasons for this change of heart by the CCPA are not disclosed, although the new element seems to be the presence of a computer in the claims. Thus it may be that the Court was concerned about the impact of denying patents for emergent technology (despite the fact that it had been to this point doing just fine without patent grants).

⁶² Robert O Nimtz, “The Patentability of Computer Programs”(1970) 1 *Rutgers Journal of Computers and Law* 38 at 40. See also Pamela Samuelson, “Benson Revisited: The Case Against Patent Protection for Algorithms and Other Computer Program-Related Inventions,” (1990) 39 *Emory Law Journal* 1025 at 1041-1048.

⁶³ Pamela Samuelson, “Benson Revisited: The Case Against Patent Protection for Algorithms and Other Computer Program-Related Inventions,” (1990) 39 *Emory Law Journal* 1025 at 1043.

⁶⁴ Pamela Samuelson, “Benson Revisited: The Case Against Patent Protection for Algorithms and Other Computer Program-Related Inventions,” (1990) 39 *Emory Law Journal* 1025 at 1044-1045.

⁶⁵ *In re Prater* (1968) 415 F.2d 1378.

⁶⁶ Pamela Samuelson, “Benson Revisited: The Case Against Patent Protection for Algorithms and Other Computer Program-Related Inventions,” (1990) 39 *Emory Law Journal* 1025 at 1045, footnote 64.

⁶⁷ *In re Prater* 415 F.2d 1378 (1968) at 1389.

⁶⁸ *In re Prater* 415 F.2d 1390 (1969).

⁶⁹ The patent was also said to fall foul of the Ninth and Tenth amendments. The Ninth Amendment states that “[t]he enumeration in the Constitution, of certain rights, shall not be construed to deny or disparage others

Chapter 3: How Software Became Patentable

upon a patentee the right to exclude others from thinking in a certain manner”.⁷⁰ The application for rehearing was dealt with on the basis that the applicants did not seek patent coverage on purely mental processes, yet the claims could be read as covering them, so therefore the Court held that the claim failed to comply with §112⁷¹ and the patent was rejected. So the argument was neither rejected nor endorsed. The argument was again argued on rehearing, although the case was decided on other grounds.⁷²

On rehearing,⁷³ the Court again avoided a direct rejection of the mental steps doctrine, although they did note a simple way of working around it:

Although in view of our decision here we find it unnecessary to analyze and/or review in depth the so-called ‘mental steps’ doctrine, it would appear that the *disclosure of apparatus for performing the process wholly without human intervention* merely shows that the disclosed process does not fall within the so-called ‘mental steps’ exclusion.⁷⁴

In the case of *Bernhart*⁷⁵, the CCPA considered the patentability of “a method of and apparatus for automatically making a two-dimensional portrayal of a three-dimensional object”.⁷⁶ The Court reiterated the traditional exclusions on patentable:

It is clear that in enacting 35 U.S.C.S. § 101, Congress meant to exclude principles or laws of nature and mathematics, of which equations are an example, from even temporary monopolization by patent. Accordingly, no rule of law should be announced which would impress a monopoly upon all uses of the equations disclosed by appellants here in their patent application.⁷⁷

However, the Court considered that the use of mathematics to describe the operation of an invention was an improper basis upon which to exclude claims since:

retained by the people”. As such it operates not as a positive source of rights, but negatively, to prevent the enumerated rights in the Constitution being used as the basis of the denial of non-enumerated rights. As such the amendment would arguably prevent the enumerated freedom of speech from being interpreted in such a way as to deny freedom of thought, on which freedom of speech depends (the relationship between freedom of thought and freedom of speech/expression is considered further in Chapter 6). The Tenth Amendment, known as the reserve powers clause states that “[t]he powers not delegated to the United States by the Constitution, nor prohibited by it to the States, are reserved to the States respectively, or to the people”. As such, the argument would seem to be that the people could not have been taken to have delegated the power to regulate their thoughts to Congress. Similar support for freedom of thought may perhaps be sourced in the “penumbra” and “emanations” of the right of free speech: see *Griswold v Connecticut* 381 US 479 (1965) per Douglas J; or as an incident of the “concept of personal ‘liberty’ embodied in the Fourteenth Amendment’s Due Process Clause”:

Roe v Wade 410 US 113 (1973) at 129. This topic will be explored in further detail in Chapter 6.

⁷⁰ *In re Prater* 415 F.2d 1390 (1969) at 1391.

⁷¹ 35 U.S.C. §112 requires that the specification contain a written description in “full, clear, concise, and exact terms”.

⁷² *In re Prater* 425 F.2d 1393 (1969) at 1400, footnote 20.

⁷³ *In re Prater* 415 F.2d 1393 (1969).

⁷⁴ *In re Prater* 415 F.2d 1393 (1969) at 1403.

⁷⁵ *In re Bernhart* 417 F.2d 1395 (1969).

⁷⁶ *In re Bernhart* 417 F.2d 1395 (1969) at 1396.

⁷⁷ *In re Bernhart* 417 F.2d 1395 (1969) at 1399.

How software became patentable subject matter

We should not penalize the inventor who makes his invention by discovering new and unobvious mathematical relationships which he then utilizes in a machine, as against the inventor who makes the same machine by trial and error and does not disclose the laws by which it operates.⁷⁸

The characterisation of the invention as physical rather than mental in nature also so their Honours focus upon the physical changes which a program, when executed on a computer, brings about:

There is one further rationale used by both the board and the examiner, namely, that the provision of new signals to be stored by the computer does not make it a new machine, i.e. it is structurally the same, no matter how new, useful and unobvious the result. This rationale really goes more to novelty than to statutory subject matter but it appears to be at the heart of the present controversy. To this question we say that *if a machine is programmed in a certain new and unobvious way, it is physically different from the machine without that program*; its memory elements are differently arranged. The fact that these physical changes are invisible to the eye should not tempt us to conclude that the machine has not been changed. If a new machine has not been invented, certainly a “new and useful improvement” of the unprogrammed machine has been, and Congress has said in 35 U.S.C. 101 that such improvements are statutory subject matter for a patent. It may well be that the vast majority of newly programmed machines are obvious to those skilled in the art and hence unpatentable under 35 U.S.C. §103. We are concluding here that such machines are statutory under 35 U.S.C. 101, and that claims defining them must be judged for patentability in light of the prior art.⁷⁹

The court also relied on the approach of the court in the *Prater* rehearing and established that the disclosure of the computer was sufficient to avoid the mental steps exclusion:

To find that the claimed process could be done mentally would require us to hold that a human mind is a digital computer or its equivalent, and that a draftsman is a planar plotting apparatus or its equivalent. On the facts of this case we are unwilling so to hold.⁸⁰

In *Musgrave*⁸¹, the Court went further, resiling from the mental process exclusion entirely in favour of a requirement that a claimed process be part of the part of the “technological arts”:

the statutory language ... contains nothing whatever which would either include or exclude claims containing “mental steps” and whatever law there may be on the subject cannot be attributed to Congress. It is purely a question of case law. That

⁷⁸ *In re Bernhart* 417 F.2d 1395 (1969) at 1399-1400.

⁷⁹ *In re Bernhart* 417 F.2d 1395 (1969) at 1400.

⁸⁰ *In re Bernhart* 417 F.2d 1395 (1969) at 1401.

⁸¹ *In re Musgrave* 431 F.2d 882 (1970).

Chapter 3: How Software Became Patentable

law we, like others, have found to be something of a morass. As indicated by footnote 22 in Judge Baldwin's *Prater* opinion, "mental" is a vague term of indefinite meaning, and whether a given step is "mental," or "purely mental," is a question which has had to be determined on a case-by-case basis, "considering all of the surrounding circumstances." Since, additionally, the legal significance of a finding that a given step was or was not "mental" or "purely mental" was itself in doubt, characterizing steps of method claims as "mental," "purely mental," "physical," or "purely physical" gave little certainty to the law.

...

We cannot agree with the board that these claims (all the steps of which can be carried out by the disclosed apparatus) are directed to non-statutory processes merely because some or all the steps therein can also be carried out in or with the aid of the human mind or because it may be necessary for one performing the processes to think. All that is necessary, in our view, to make a sequence of operational steps a statutory "process" within 35 USC 101 is *that it be in the technological arts* so as to be in consonance with the Constitutional purpose to promote the progress of "useful arts."⁸²

In the case of *Benson*,⁸³ the applicants, on behalf of AT&T, applied for a patent on "a method for converting binary coded-decimal (BCD) numerals into pure binary numerals ... in a general-purpose digital computer of any type."⁸⁴ The claims were rejected by the examiner as not being directed to patentable subject matter on the grounds that "they set forth 'mental processes' and 'mathematical steps,' neither being an 'art' as construed by a long line of decisions".⁸⁵ The CCPA overturned the decision,⁸⁶ and were scathing in their attack on the Patent Office, whose persistent adherence to the mental process doctrine, despite the CCPA's adverse treatment of it, was clearly a source of frustration for the Court:

The reasoning of the board's opinion bears a remarkable similarity to the opinion of the board in *Musgrave*, which is not surprising since it was the same board and the opinion was by the same member. The same line of cases is relied on here that was relied on in *Prater* and *Musgrave* and we have given them full consideration before, particularly in those two cases in which we considered at length the 'mental steps' doctrine.⁸⁷

Despite these similarities, the Court had earlier noted the instant claims contained a "significant difference".⁸⁸ The claims here were "directed solely to the art of data-processing itself whereas

⁸² *In re Musgrave* 431 F.2d 882 (1970) at 890-893.

⁸³ *In re Benson* 441 F.2d 682 (1971).

⁸⁴ *Gottschalk v Benson* 409 US 63 (1973) at 64.

⁸⁵ Cited in *In re Benson* 441 F.2d 682 (C.C.P.A. 1971).

⁸⁶ *In re Benson* 441 F.2d 682 (C.C.P.A. 1971). The decision of the court was written by Rich J, who had strenuously dissented on the *Prater* petition for rehearing, and whose dogged dedication to a broad understanding of subject matter patentability had not waned by 2001 when he wrote the judgment in *State Street* (discussed below).

⁸⁷ *In re Benson* 441 F.2d 682 (C.C.P.A. 1971) at 686-687.

⁸⁸ *In re Benson* 441 F.2d 682 (C.C.P.A. 1971) at 686.

How software became patentable subject matter

in most of the [previous] cases some subsidiary or additional art was involved.”⁸⁹ However the Court went on to characterise the claims as “a typical product of its time”,⁹⁰ since it, like its predecessors, represented “the outgrowth of a blanket Patent Office policy ... to deny claims such as those before us here on the ground that they were not statutory subject matter.”⁹¹

The question Judge Baldwin asked in *Musgrave* is the question we have here: “would a reasonable interpretation of the claims include coverage of the process implemented by the human mind?” The answer clearly is “No”.⁹²

In relation to the first disputed claim (claim 8) the Court noted that the method claimed was linked to computer hardware, namely a “shift register”,⁹³ and that it operated on “signals”.⁹⁴ As such the claim covered “only a machine-implemented process... the process can be carried out with no intervention by a human being once the apparatus is set up”.⁹⁵ This being the case, their Honours were unpersuaded with the argument of the solicitor that

the method is not a “process” within section 101 on the ground that a programmable computer is merely a “tool of the mind” and the method is basically “mental” in character, apparently because the “workstuff” of the method is numbers which are mathematical abstractions.⁹⁶

The Court rejected this suggestion on the basis that other machines such as “[c]ash registers, bookkeeping machines and adding machines also work only with numbers but this has never been considered a ground for [considering them non-patentable].”⁹⁷ With respect to the Court, who were undoubtedly dealing with unfamiliar and complex technical concepts, such an analogy is very poor. As Samuelson notes,

[t]he Patent Office did not argue that computers or cash registers would be unpatentable as machines merely because they could be used to add or subtract numbers. The Office argued that if the *process* that both cash registers and computers could carry out ... could also be carried out in a person’s head, *that* process would be unpatentable as a mental process.⁹⁸

Although “no reference to any apparatus”⁹⁹ was expressly disclosed in the second disputed claim, claim 13, the Court held that the claim was limited to the “identical programmed sup-

⁸⁹ *In re Benson* 441 F.2d 682 (C.C.P.A. 1971) at 686.

⁹⁰ *In re Benson* 441 F.2d 682 (C.C.P.A. 1971) at 686.

⁹¹ *In re Benson* 441 F.2d 682 (C.C.P.A. 1971) at 686.

⁹² *In re Benson* 441 F.2d 682 (C.C.P.A. 1971) at 687.

⁹³ See *In re Benson* 441 F.2d 682 (C.C.P.A. 1971) at 687: “a hardware element constructed so as to perform the shifting of its contained data”.

⁹⁴ *In re Benson* 441 F.2d 682 (C.C.P.A. 1971) at 687: “[this] can only mean signals of the kind upon which the disclosed electronic digital computer hardware operates”.

⁹⁵ This is a clear reference to *In re Prater* 415 F.2d 1393 (C.C.P.A. 1969) at 1403.

⁹⁶ *In re Benson* 441 F.2d 682 (C.C.P.A., 1971) at 687.

⁹⁷ *In re Benson* 441 F.2d 682 (C.C.P.A. 1971) at 687.

⁹⁸ Pamela Samuelson, “Benson Revisited: The Case Against Patent Protection for Algorithms and Other Computer Program-Related Inventions,” (1990) 39 *Emory Law Journal* 1025 at 1109.

⁹⁹ *In re Benson* 441 F.2d 682 (C.C.P.A. 1971) at 687.

Chapter 3: How Software Became Patentable

ports claim 8"¹⁰⁰ and was similarly directed to patentable subject matter. However, the court noted that in the absence of a requirement that such a computer be used, "the method within the claim can be practiced either with apparatus other than that described or with the simplest of equipment... This could in theory be any kind of writing implement and any kind of recording medium - 'pencil and paper'".¹⁰¹

Given the clear conflict with the Patent Office interpretation of the mental steps doctrine, it is interesting to note the apparent concession made to the possibility of mental steps continuing to exist in the *Abrams* sense:

Only in the manual performance would it require the operator even to think and then only to the extent necessary to assure that he is doing what the claim tells him to do. In no case is the exercise of judgment required or even the making of a decision as between alternatives.¹⁰²

The Court then went on to note that in any event, the method claimed "has no practical use other than the more effective operation of a machine known as a digital computer".¹⁰³ As such, since these "are in the technological field, are a part of our best-known technologies"¹⁰⁴ they were therefore "in the 'useful arts' rather than the 'liberal arts' as are all other types of 'business machines'... How can it be said that a process having no practical value other than enhancing the internal operation of those machines is not likewise in the technological or useful arts?"¹⁰⁵

On the approach advocated by the CCPA, the doctrine of mental steps would have little further application in the patentable subject matter inquiry, since all that is required to avoid it is the recitation of a "link" to some form of apparatus, namely a computer, to avoid its application. As such, the doctrine was reduced to a mere form requirement. The Patent Office, still unwilling to back down on the issue, appealed the decision to the Supreme Court.

Gottschalk v Benson

In *Gottschalk v Benson*,¹⁰⁶ the US Supreme Court ruled for the first time on the patentability of a software-related invention, and in doing so set the tone for the Court's view on the patentability of computer programs. Although the Court was subsequently criticised for failing to adequately enunciate the rationale for its decision,¹⁰⁷ one thing that can be said with certainty

¹⁰⁰ *In re Benson* 441 F.2d 682 (C.C.P.A. 1971) at 687.

¹⁰¹ *In re Benson* 441 F.2d 682 (C.C.P.A. 1971) at 688.

¹⁰² *In re Benson* 441 F.2d 682 (C.C.P.A. 1971) at 688.

¹⁰³ *In re Benson* 441 F.2d 682 (C.C.P.A. 1971) at 688.

¹⁰⁴ *In re Benson* 441 F.2d 682 (C.C.P.A. 1971) at 688.

¹⁰⁵ *In re Benson* 441 F.2d 682 (C.C.P.A. 1971) at 688.

¹⁰⁶ *Gottschalk v Benson* 409 US 63 (1973).

¹⁰⁷ Rich J, who was involved in so many of the mental steps cases subsequently noted in *In re Johnston* (1974) 502 F.2d 765 at 773-774: "I am probably as much - if not more - confused by the wording of the Benson opinion as many others. ... I have no idea what was in the collective mind of the six-justice Court...". Chisum similarly disparages the decision as "monstrously bad": Donald S. Chisum, "The Patentability of Algorithms," (1986) 47 *University of Pittsburgh Law Review* 959. Samuelson more generously notes that the "decision is not a model of clarity": Pamela Samuelson, "Benson Revisited: The Case Against Patent Protection for Algorithms and Other Computer Program-Related Inventions," (1990) 39 *Emory Law Journal* 1025.

How software became patentable subject matter

is that it did not endorse the approach of the CCPA. Rather than tackle the mental steps doctrine however, the Court picked up on the argument of the appellant before the CCPA that the claims were “directed to a machine *algorithm*”¹⁰⁸ and not patent eligible on that basis. Thus rather than directly addressing the propriety of the mental steps exclusion, the Supreme Court redirected attention to a new issue - whether *algorithms* should be patentable.

Samuelson notes that subsequent to this case, all reference to mental steps in case law all but disappears.¹⁰⁹ It may be that, as Newell suggests, “[s]equences of mental steps and algorithms are the same thing”.¹¹⁰ It is submitted however that the net effect of this change of direction has been to steer the issue off course - the subsequent developments which have eroded *Benson* were to be expected, given the divergence of opinion between the Supreme Court, lower courts and the USPTO. The use of the term “algorithm” encouraged attempts to distinguish between mathematical and non-mathematical “algorithms” – a distinction which is entirely illusory. As Newell points out:

mathematics deals with both nonnumerical things and numerical things. Correspondingly, there are both numerical and nonnumerical algorithms. Therefore, any attempt to find a helpful or cutting distinction between mathematics and nonmathematics, as between numerical or nonnumerical, is doomed. Indeed, in the mid-1930s the central argument of a famous proof, by the mathematician Kurt Gödel, involved showing that all of logic is a part of number theory. The scheme, which ever after has been called Gödel numbering, assigns an integer to each logical expression, such that all the truths in logic become theorems about the integers. Although lawyers need never become acquainted with Gödel numbering, they should realize there is an underlying identity between the numerical and the nonnumerical realms that will confound any attempt to create a useful distinction between them.¹¹¹

The Court suggested a number of reasons that the claimed invention was not patentable subject matter. One argument made by the court was that “mathematical innovations should be treated like scientific truths and laws of nature”¹¹² which are inherently non-patentable subject matter. Later in the judgement, the Court emphasised that the claim was “so abstract and sweeping as to cover both known and unknown uses,”¹¹³ thus hindering the “onward march of science”¹¹⁴ through independent invention. The court also posited that the practical effect of allowing such

¹⁰⁸ *In re Benson* 441 F.2d 682 (C.C.P.A. 1971) at 682 (footnote 2).

¹⁰⁹ Pamela Samuelson, “Benson Revisited: The Case Against Patent Protection for Algorithms and Other Computer Program-Related Inventions,” (1990) 39 *Emory Law Journal* 1025 at 1042-1043.

¹¹⁰ Allen Newell, “Response: The Models Are Broken, The Models Are Broken,” (1986) 47 *University of Pittsburgh Law Review* 1023 at 1025.

¹¹¹ Allen Newell, “Response: The Models Are Broken, The Models Are Broken,” (1986) 47 *University of Pittsburgh Law Review* 1023 at 1024-1025. Gödel numbering is the key step in the proof of his Incompleteness Theorem, something which is discussed in much greater detail later in this thesis.

¹¹² Pamela Samuelson, “Benson Revisited: The Case Against Patent Protection for Algorithms and Other Computer Program-Related Inventions,” (1990) 39 *Emory Law Journal* 1025 at 1053. *Gottschalk v Benson*, 409 US 63 (1973) at 67.

¹¹³ *Gottschalk v Benson* 409 US 63 (1973) at 68.

¹¹⁴ *Gottschalk v Benson* 409 US 63 (1973) at 68, citing *O'Reilly v Morse* 56 US 62 (1852).

Chapter 3: How Software Became Patentable

a claim, in the absence of a specific practical application, would be to award a patent for an idea – clearly non-patentable subject matter.¹¹⁵

Parker v Flook

In *Flook*,¹¹⁶ the Supreme Court considered the patentability of a method for updating alarm limits during a catalytic conversion process. Whereas the claimed invention in *Benson* subsisted in an algorithm alone, the invention in *Flook* claimed a mathematical algorithm as one step in the method. However, “[t]he only difference between the conventional methods of changing alarm limits and that described in [the] respondent’s application rests in ... the mathematical algorithm or formula”.¹¹⁷ That is, the only advance over the prior art resides in the mathematical algorithm. Thus the court characterised the central issue in this case as “whether the identification of a limited category of useful, though conventional, post-solution applications of such a formula makes respondent’s method eligible for patent protection”.¹¹⁸

The line between a patentable “process” and an unpatentable “principle” is not always clear. Both are “[conceptions] of the mind, seen only by [their] effects when being executed or performed.” *Tilghman v Proctor* 102 US 707, 728¹¹⁹

In the end, the court found the claimed invention to be as non-patentable as the pure algorithm claimed in *Benson*. Their reasoning was that

[t]he notion that post-solution activity, no matter how conventional or obvious in itself, can transform an non-patentable principle into a patentable process exalts form over substance. A competent draftsman could attach some form of post-solution activity to almost any mathematical formula.¹²⁰

Despite this, however, the judgement in *Flook* can be taken to have “backing down” from *Benson* in that the court expressly held the question of patentability open as follows:

Neither the dearth of precedent, nor this decision, should ... be interpreted as reflecting a judgement that patent protection of certain novel and useful computer programs will not promote the progress of science and the useful arts, or that such protection is undesirable as a matter of policy.¹²¹

The court also point out that it is not the mere existence of an algorithm in the claimed process which defeated the patentability of the invention. It was the fact that “once the algorithm is assumed to be within the prior art, the application, considered as a whole, contains no patentable invention”.¹²²

¹¹⁵ *Gottschalk v Benson* 409 US 63 (1973) at 71.

¹¹⁶ *Parker v Flook* 437 US 584 (1978).

¹¹⁷ *Parker v Flook* 437 US 584 (1978) at 586.

¹¹⁸ *Parker v Flook* 437 US 584 (1978) at 585.

¹¹⁹ *Parker v Flook* 437 US 584 (1978) at 589.

¹²⁰ *Parker v Flook* 437 US 584 (1978) at 590.

¹²¹ *Parker v Flook* 437 US 584 (1978) at 595.

¹²² *Parker v Flook* 437 US 584 (1978) at 594.

In re Freeman & In re Walter

In between the Supreme Court decisions in *Benson* and *Flook* and the later decision in *Diehr*, the CCPA developed a two-step test to determine whether claimed inventions containing algorithms should be considered patentable subject matter or not.

In *Freeman*,¹²³ the court dealt with the patentability of “a system for typesetting alphanumeric information, using a computer-based control system in conjunction with a phototypesetter of conventional design,”¹²⁴ a system which was “especially useful in printing mathematical formulae”.¹²⁵ In determining whether *Benson* precluded patentability, the court resolved that a two-step analysis was required:

First, it must be determined whether the claim directly or indirectly recites an ‘algorithm’ in the *Benson* sense of that term, for a claim which fails even to recite an algorithm clearly cannot wholly pre-empt an algorithm. Second, the claim must be further analyzed to determine whether in its entirety it wholly pre-empts that algorithm.¹²⁶

In discussing the first step, the court took a narrow view of the concept of an algorithm, requiring that the claims “recite process steps which are themselves mathematical calculations, formulae or equations”.¹²⁷ The court discussed, but rejected, a broader definition, namely “a step-by-step procedure for solving a problem or accomplishing some end”¹²⁸ on the basis that “step-by-step solutions often attain the status of patentable inventions”.¹²⁹ Since the claimed invention did not involve any mathematical algorithms, it was held to fail the first step of the stated test, and hence to be patentable subject matter.

The main significance of *Walter*¹³⁰ was that the CCPA modified the second step of the *Freeman* test, moving away from the issue of pre-emption of an algorithm,¹³¹ and focusing on the limiting nature of the physical aspects of the claimed invention. The court restated the second test as follows:

If it appears that the mathematical algorithm is implemented in a specific manner to define structural relationships between the physical elements of the claim (in apparatus claims) or to refine or limit claim steps (in process claims), the claim passes muster under §101. If, however, the mathematical algorithm is merely presented and solved by the claimed invention, as was the case in *Benson* and *Flook*, and is not applied in any manner to physical elements or process steps, no amount of

¹²³ *In re Freeman* 573 F.2d 1243 (1978).

¹²⁴ *In re Freeman* 573 F.2d 1243 (1978) at 1238.

¹²⁵ *In re Freeman* 573 F.2d 1243 (1978) at 1239.

¹²⁶ *In re Freeman* 573 F.2d 1243 (1978) at 1245.

¹²⁷ *In re Freeman* 573 F.2d 1243 (1978) at 1246.

¹²⁸ *In re Freeman* 573 F.2d 1243 (1978) at 1245.

¹²⁹ *In re Freeman* 573 F.2d 1243 (1978) at 1245.

¹³⁰ *In re Walter* 618 F.2d 766 (1980).

¹³¹ *In re Walter* 618 F.2d 766 (1980) at 767.

Chapter 3: How Software Became Patentable

post-solution activity will render the claim statutory; nor is it saved by a preamble merely reciting the field of use of the mathematical algorithm.¹³²

Diamond v Chakrabarty

In *Chakrabarty*,¹³³ the Supreme Court held that a “live, human-made micro-organism”¹³⁴ could be patentable. In doing so, the Court adopted a wide definition of patentable subject matter, holding that “Congress intended statutory subject matter to ‘include anything under the sun that is made by man.’”¹³⁵ On first glance this may seem to remove all limits on subject matter entirely. It certainly seems to involve a more expansive definition than even the Australian High Court case of *NRDC*,¹³⁶ being roughly analogous to an “artificially created state of affairs”, without any requirement of “value in the field of economic endeavour. In thus holding, Burger CJ relied on the legislative history of patent law in the US, in particular relying on the notion that “[t]he Act embodied Jefferson’s philosophy that ‘ingenuity should receive a liberal encouragement.’”¹³⁷ Furthermore the court noted that “[i]n choosing such expansive terms as ‘manufacture’ and ‘composition of matter,’ modified by the comprehensive ‘any,’ Congress plainly contemplated that the patent laws would be given wide scope.”¹³⁸

The reference to the phrase “anything under the sun” is often cited in support of a broad approach to patentability.¹³⁹ However,

that phrase is a misleading quotation from the legislative history of the Patent Act of 1952. The full quote clearly acknowledges the statutory limitations to patentable subject matter: “A person may have invented a machine or a manufacture, which may include anything under the sun made by man, but it is not necessarily patentable under section 101 unless the conditions of the title are fulfilled.” H.R. Rep. No. 1923, 82d Cong., 2d Sess. 6 (1952).¹⁴⁰

In fact the phrase would be entirely consistent with a physicality constraint of the kind set out in cases earlier discussed, since only an object with a physical existence could in any real sense be said to exist “under the sun”. This position is consistent with the context in which the statement is made, namely in relation to *machines* and *manufactures*, which are two of the four categories of patentable subject matter in the *Patents Act 1952* (US).¹⁴¹ These two classes,

¹³² *In re Walter* 618 F.2d 766 (1980) at 767.

¹³³ *Diamond v Chakrabarty* 447 U.S. 303 (1980).

¹³⁴ *Diamond v Chakrabarty* 447 U.S. 303 (1980) at 305.

¹³⁵ *Diamond v Chakrabarty* 447 US 303 (1980) at 309.

¹³⁶ Discussed in detail in Chapter 2, Section 6 on page 61.

¹³⁷ *Diamond v Chakrabarty* 447 US 303 (1980) at 308, citing 5 *Writings of Thomas Jefferson* 75-76 (Washington ed. 1871).

¹³⁸ *Diamond v Chakrabarty* 447 US 303 (1980) at 308.

¹³⁹ See *Bilski v Kappos* 130 S. Ct. 3218 (2010) per Kennedy J, writing for the majority (the case is discussed in detail below).

¹⁴⁰ *Association for Molecular Pathology et al v USPTO et al* 9 Civ 4515 (US District Court, Southern District of New York, 2010) at 110, footnote 43 per Sweet J. Similarly, the Court in *In re Warmerdam*, 33 F.3d 1354, 1358 (Fed Cir, 1994) noted that “[t]o include some things [in § 101] is to exclude others.” See also *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3237 per Stevens J (discussed below).

¹⁴¹ The other two categories are compositions of matter, and processes.

How software became patentable subject matter

as with the composition of matter the subject of the instant claims,¹⁴² all relate to inventions which clearly have physical manifestations. It is only the category of processes which might be envisaged as being able to be interpreted broadly, without reference to any corporeal embodiment. It is submitted however that to do so would be to go beyond a reasonable interpretation of the statement.¹⁴³

In any event, Burger CJ did acknowledge that some things remained non-patentable, namely “laws of nature, physical phenomena, and abstract ideas”.¹⁴⁴ According to his Honour, a micro-organism was “not to a hitherto unknown natural phenomenon, but to a non-naturally occurring manufacture or composition of matter – a product of human ingenuity.”¹⁴⁵

Diamond v Diehr

*Diehr*¹⁴⁶ was the final word of the US Supreme Court on the issue of software patenting until very recently.¹⁴⁷ The invention in *Diehr* was a process for moulding and curing rubber in which a computer iteratively applied a well-known equation to determine the optimum time at which to remove the rubber from the molding press. The constant application of the equation improved the process by taking into account the fluctuating temperature of the rubber inside the mold.

The decision in *Diehr* changed the focus of the inquiry yet again. Rather than focusing on the importance of the algorithm to the claimed invention, the Supreme Court stressed the importance of the industrial nature of the claimed process – something traditionally within the limits of patentability:

[W]hen a claim containing a mathematical formula implements or applies that formula in a structure or process which, when considered as a whole, is performing a function which the patent laws were designed to protect (for example, transforming or reducing an article to a different state or thing), then the claim satisfies the requirements of [patentability].¹⁴⁸

The majority rejected the ‘point of novelty’ approach of *Flook*, claiming that “[i]t is inappropriate to dissect the claims into old and new elements and then to ignore the presence of the old elements in the analysis.”¹⁴⁹ Thus, the “question ... of whether a particular invention is novel is wholly apart from whether the invention falls into a category of statutory subject matter”.¹⁵⁰

¹⁴² See below.

¹⁴³ See *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3237 per Stevens J.

¹⁴⁴ *Diamond v Chakrabarty* 447 U.S. 303 (1980) at 309.

¹⁴⁵ *Diamond v Chakrabarty* 447 U.S. 303 (1980) at 309.

¹⁴⁶ *Diamond v Diehr* 450 US 175 (1981).

¹⁴⁷ The Supreme Court did not address the issue again for nearly 20 years, until the decision in *Bilski v Kappos* 130 S. Ct. 3218 (2010), which is discussed in detail below.

¹⁴⁸ *Diamond v Diehr* 450 US 175 (1981) at 192.

¹⁴⁹ *Diamond v Diehr* 450 US 175 (1981) at 188.

¹⁵⁰ *Diamond v Diehr* 450 US 175 (1981) at 190.

Chapter 3: How Software Became Patentable

Despite the different focus, the majority clearly reaffirmed that mathematical inventions were outside the limits of patentability, drawing a distinction between a non-patentable discovery of a mathematical equation on the one hand, and its “application” on the other.¹⁵¹

Diehr has subsequently been treated as opening the door to the patenting of software. However, “the *Diehr* decision was regarded as a very limited one for many years. It is limited in that it affirms only that patents can issue for traditionally patentable industrial processes which include a computer program as an element.”¹⁵² In other words, the decision was *not* regarded at the time at which it was decided, as consistent with the patenting of “pure” software.

Such a view also takes account of the strong dissent delivered by a 4-member minority, in which they attacked the improper characterisation of the invention by the majority, rightly pointing out that “[t]he starting point in the proper adjudication of patent litigation is an understanding of what the inventor claims to have discovered”.¹⁵³ The difference in characterisation between the minority and majority is outlined by Stevens J as follows:

As the Court reads the claims in the *Diehr* and *Lutton* patent application, the inventors’ discovery is a method of constantly measuring the actual temperature inside a rubber molding press. As I read the claims, their discovery is an improved method of calculating the time that the mold should remain closed during the curing process. If the Courts reading of the claims were correct, I would agree that they disclose patentable subject matter. On the other hand, if the Court accepted my reading, I feel confident that the case would be decided differently.¹⁵⁴

The minority characterisation is consistent with the ‘point of novelty’ approach taken in *Flook*. In fact, as Stevens J points out, there is a great deal of similarity between the *Diehr* and *Flook* claims in that “[t]he essence of the claimed discovery in both cases was an algorithm that could be programmed on a digital computer.”¹⁵⁵

The minority judgement includes a frank admission about the difficulties of determining the issue of computer program patentability:

The broad question of whether computer programs should be given patent protection involves policy considerations that this Court is not authorized to address ... As the numerous briefs amicus curiae filed in [*Benson*, *Flook*, *Dann v Johnston*] and this case demonstrate, that question is not only difficult and important, but apparently also one that may be affected by institutional bias. In each of those cases, the spokesmen for the organized patent bar have uniformly favored patentability and industry representatives have taken positions properly motivated by their economic self-interest. Notwithstanding fervent argument that patent protection is essential

¹⁵¹ “It is now commonplace that an application of a law of nature or mathematical formula to a known structure or process may well be deserving of patent protection. *Diamond v Diehr* 450 US 175 (1981) at 187.

¹⁵² Pamela Samuelson, “*Benson* Revisited: The Case Against Patent Protection for Algorithms and Other Computer Program-Related Inventions,” (1990) 39 *Emory Law Journal* 1025 at 1029.

¹⁵³ *Diamond v Diehr* 450 US 175 (1981) at 193-194.

¹⁵⁴ *Diamond v Diehr* 450 US 175 (1981) at 206-207. (footnotes omitted)

¹⁵⁵ *Diamond v Diehr* 450 US 175 (1981) at 209.

How software became patentable subject matter

for the growth of the software industry, commentators have noted that ‘this industry is growing by leaps and bounds without it.’ In addition, even some commentators who believe that legal protection for computer programs is desirable have expressed doubts that the present patent system can provide the needed protection.¹⁵⁶

In re Abele

Eventually the Federal Circuit in *Abele*¹⁵⁷ took a view of *Diehr* which opened the door to the unfettered patentability of software. In *Abele*, whilst discussing a claimed “improvement in computed tomography”,¹⁵⁸ the court modified the *Walter* restatement of the second limb of the *Freeman* test as follows:

Walter should be read as requiring no more than that the algorithm be ‘applied in any manner to physical elements or process steps,’ provided that its application is circumscribed by more than a field of use limitation or non-essential post-solution activity. Thus if the claim would be ‘otherwise statutory’ albeit less useful without the algorithm, the claim likewise presents statutory subject matter when the algorithm is included.¹⁵⁹

This restatement effectively sought to modify the *Freeman-Walter* test in line with the principles set out in *Diehr*. The resulting *Freeman-Walter-Abele* test was adopted by the Australian Patent Office in 1986¹⁶⁰ and continued to be the test for the patentability of software up until it was rejected by Burchett J in *IBM*.¹⁶¹

Following *In re Abele*, the *Freeman-Walter-Abele* (FWA) test required the following approach to determining patentability:

1. Determine whether the claim recites an algorithm within the meaning of *Benson*.
2. Determine whether the algorithm is applied in any manner to physical elements or process steps. If it is, then it is patentable subject matter.

Alappat

In *Alappat*,¹⁶² the Court in considering the patentability of “means for creating a smooth waveform display in a digital oscilloscope”,¹⁶³ abandoned the FWA test in favour of a “whole invention” approach. On such an approach, the FWA test was redundant, since “even in those

¹⁵⁶ *Diamond v Diehr* 450 US 175 (1981) at 216-217.

¹⁵⁷ *In re Abele* 684 F.2d 902 (1982).

¹⁵⁸ *In re Abele* 684 F.2d 902 (1982) at 903.

¹⁵⁹ *In re Abele* 684 F.2d 902 (1982) at 907.

¹⁶⁰ Australian Patent Office, *Guidelines for Considering the Patentability of Computer Program Related Inventions* (Canberra, March 1986).

¹⁶¹ *IBM v Commissioner of Patents* (1991) 33 FCR 218, discussed at 3 on page 121 below.

¹⁶² *In re Alappat* 33 F.3d 1526 (1994).

¹⁶³ *In re Alappat* 33 F.3d 1526 (1994) at 1537.

Chapter 3: How Software Became Patentable

cases wherein courts have applied a variant of the two-part analysis of [*Freeman* and *Walter*], the ultimate issue always has been whether the claim as a whole is drawn to statutory subject matter.”¹⁶⁴ As such,

the proper inquiry in dealing with the so called mathematical subject matter exception to § 101 alleged herein is to see whether the claimed subject matter as a whole is a disembodied mathematical concept, whether categorized as a mathematical formula, mathematical equation, mathematical algorithm, or the like, which in essence represents nothing more than a “law of nature,” “natural phenomenon,” or “abstract idea.” If so, *Diehr* precludes the patenting of that subject matter.

Applying this approach to the invention at hand, the Court held that “[this was] not a disembodied mathematical concept which may be characterized as an ‘abstract idea,’ but rather a specific machine to produce a *useful, concrete, and tangible result*.”¹⁶⁵ The Court also relied on its earlier line of reasoning from *Musgrave* and *Bernhart* “that such programming creates a new machine, because a general purpose computer in effect becomes a special purpose computer once it is programmed to perform particular functions pursuant to instructions from program software”.¹⁶⁶ As such, “a computer, like a rasterizer, is apparatus not mathematics”.¹⁶⁷ Such a result was in the Court’s view consistent with the fact that “[t]he Supreme Court has never held that a programmed computer may never be entitled to patent protection”.¹⁶⁸

State Street

The Federal Circuit Court of Appeal in *State Street*,¹⁶⁹ a decision authored by Rich J, reversed the District Court’s rejection of a claim for a “Data Processing System for Hub and Spoke Financial Services Configuration”,¹⁷⁰ or more specifically “a system that allows an administrator to monitor and record the financial information flow and make all calculations necessary for maintaining a partner fund financial services configuration.”¹⁷¹

The case illustrates the importance of characterisation to the subject matter inquiry. The District Court, who had granted summary judgment on the basis of improper subject matter, looked past the way in which the system had been drafted as apparatus claims, noting:

[i]n all instances, this critical question must be answered: ‘What did applicants invent?’” *Grams*, 888 F.2d at 839 (quoting *Abele*, 684 F.2d 902, 907). At bottom, the invention is an accounting system for a certain type of financial investment vehicle claimed as means for performing a series of mathematical functions. Quite simply,

¹⁶⁴ *In re Alappat* 33 F.3d 1526 (1994) at 1544, footnote 21.

¹⁶⁵ *In re Alappat* 33 F.3d 1526 (Fed Cir, 1994) at 1544 (emphasis added).

¹⁶⁶ *In re Alappat* 33 F.3d 1526 (1994) at 1545.

¹⁶⁷ *In re Alappat* 33 F.3d 1526 (1994) at 1545.

¹⁶⁸ *In re Alappat* 33 F.3d 1526 (1994) at 1545.

¹⁶⁹ *State Street Bank & Trust Co v Signature Financial Group Inc* 149 F.3d 1368 (1998).

¹⁷⁰ *State Street Bank & Trust Co v Signature Financial Group Inc* (1998) 149 F.3d 1368 at 1370.

¹⁷¹ *State Street Bank & Trust Co v Signature Financial Group Inc* 149 F.3d 1368 (1998) at 1371.

How software became patentable subject matter

it involves no further physical transformation or reduction than inputting numbers, calculating numbers, outputting numbers, and storing numbers.¹⁷²

Such a system obviously ran afoul of the “mathematical algorithm exception”¹⁷³ as set out in *Benson*. Similarly, the District Court held that “[p]atenting an accounting system necessary to carry on a certain type of business is tantamount to a patent on the business itself. Such abstract ideas are not patentable, either as methods of doing business or as mathematical algorithms.”¹⁷⁴

The Federal Circuit however noted that the original application contained six process claims and six machine claims, the former being cancelled when the examiner “contemplated a [subject matter] rejection”. As such, the Court “looked no further than the applicant’s apparatus claim and determined that the subject matter was a machine, and therefore, statutory”.¹⁷⁵ In order to demonstrate that fact, Rich J took the unusual step of rewriting the claims to emphasize the computer hardware by which the method would be performed.¹⁷⁶

In relation to the mathematics exception, the Court held that the FWA test was rendered nugatory by the more expansive approach to inherent patentability found in *Diehr*, *Chakrabarty* and *Alappat*.¹⁷⁷ Rich J noted that “[u]npatentable mathematical algorithms are identifiable by showing that they are merely abstract ideas constituting disembodied concepts or truths that are not ‘useful’.”¹⁷⁸ However, this would not be so where the subject matter was “reduced to some type of practical application, i.e., ‘a useful, concrete and tangible result’.”¹⁷⁹ As such, the presently claimed invention was patentable subject matter since

the transformation of data, representing discrete dollar amounts, by a machine through a series of mathematical calculations into a final share price, constitutes a practical application of a mathematical algorithm, formula, or calculation, because it produces “a useful, concrete and tangible result”—a final share price momentarily fixed for recording and reporting purposes...¹⁸⁰

This was a step forward over the previous requirement in *Alappat* that the data being transformed represents a measurable physical phenomenon.

Rich J also “forcibly wiped out”¹⁸¹ the business method exception:

We take this opportunity to lay this ill-conceived exception to rest. Since its inception, the “business method” exception has merely represented the application of some general, but no longer applicable legal principle, perhaps arising out of the

¹⁷² *State Street Bank & Trust Co v Signature Financial Group Inc* 927 F. Supp. 502 (D Mass, 1996) at 515.

¹⁷³ *State Street Bank & Trust Co v Signature Financial Group Inc* 149 F.3d 1368 (1998) at 1372.

¹⁷⁴ *State Street Bank & Trust Co v Signature Financial Group Inc* 927 F. Supp. 502 (D Mass, 1996) at 516.

¹⁷⁵ Gregory Stobbs, *Software Patents* (2nd ed, Aspen, 2000) at 183-184.

¹⁷⁶ *State Street Bank & Trust Co v Signature Financial Group Inc* 149 F.3d 1368 (1998) at 1371-1372.

¹⁷⁷ See *State Street Bank & Trust Co v Signature Financial Group Inc* (1998) at 1373-1374.

¹⁷⁸ *State Street Bank & Trust Co v Signature Financial Group Inc* (1998) at 1373.

¹⁷⁹ *State Street Bank & Trust Co v Signature Financial Group Inc* 149 F.3d 1368 (1998) at 1373.

¹⁸⁰ *State Street Bank & Trust Co v Signature Financial Group Inc* 149 F.3d 1368 (1998) at 1373.

¹⁸¹ Gregory Stobbs, *Software Patents* (2nd ed, Aspen, 2000) at 184.

Chapter 3: How Software Became Patentable

“requirement for invention”—which was eliminated by §103. Since the 1952 Patent Act, business methods have been, and should have been, subject to the same legal requirements for patentability as applied to any other process or method.¹⁸²

As to whether business methods were generally non-patentable as “abstract ideas” as suggested by the District Court, his Honour remained silent.

Ex parte Lundgren

In the wake of *State Street*, it was thought that so-called “pure” business methods¹⁸³ were still non-patentable, with all business method inventions being thought to require a technological basis.¹⁸⁴ This requirement was eventually removed by the USPTO Board of Appeals in *Ex parte Lundgren*.¹⁸⁵

In that case, the BPAI considered the patentability of claims to “a method of compensating a manager ... of a privately owned primary firm for the purpose of reducing ... incentives for industry collusion”.¹⁸⁶ The examiner had rejected the claims on the basis that “the invention and the practical application to which it is directed [are] outside the technological arts, namely an economic theory expressed as a mathematical algorithm without the disclosure or suggestion of computer, automated means, [or] apparatus of any kind”.¹⁸⁷ The examiner had however withdrawn a rejection on the basis that the claims “failed to produce a useful, concrete and tangible result”,¹⁸⁸ from which the Board inferred that this *State Street* requirement had been fulfilled.

A majority of the Board held that despite what was said in *Musgrave*, “there is currently no judicially recognized ‘technological arts’ test to determine patent eligible subject matter under § 101. We decline to create one. Therefore, it is apparent that the examiner’s rejection can not be sustained.”¹⁸⁹

Judge Barrett agreed with the majority that no separate “technological arts” requirement existed,¹⁹⁰ but disagreed with the apparent inference from the judgment that any “claim to a series of steps” was a patentable process.¹⁹¹ Instead after a lengthy analysis of the historical basis of the patentable subject matter inquiry, he concluded that “all cases where statutory subject matter was found can be explained”¹⁹² by defining process to require:

¹⁸² *State Street Bank & Trust Co v Signature Financial Group Inc* 149 F.3d 1368 (1998) at 1375.

¹⁸³ Those business methods claimed in the absence of a technological means, such as a computer, to carry them out.

¹⁸⁴ *Ex parte Bowman* 61 USPQ 1669 (2001). The origin of such a requirement can be found *In re Musgrave* 57 CCPA 1352 (1970); *In re Toma* 575 F.2d 872 (1978).

¹⁸⁵ *Ex parte Lundgren* Appeal No 2003-2088 (BPAI, 2005).

¹⁸⁶ *Ex parte Lundgren* Appeal No 2003-2088 (BPAI, 2005) at 1.

¹⁸⁷ *Ex parte Lundgren* Appeal No 2003-2088 (BPAI, 2005) at 4, citing the examiner, Paper No 60, page 7.

¹⁸⁸ *Ex parte Lundgren* Appeal No 2003-2088 (BPAI, 2005) at 5, citing the examiner, Paper No 60.

¹⁸⁹ *Ex parte Lundgren* Appeal No 2003-2088 (BPAI, 2005) at 9.

¹⁹⁰ *Ex parte Lundgren* Appeal No 2003-2088 (BPAI, 2005) at 16.

¹⁹¹ *Ex parte Lundgren* Appeal No 2003-2088 (BPAI, 2005) at 16.

¹⁹² *Ex parte Lundgren* Appeal No 2003-2088 (BPAI, 2005) at 87.

How software became patentable subject matter

a transformation of physical subject matter to a different state or thing. ... [T]he subject matter transformed can be tangible or intangible, which I interpret to be matter or some form of energy to be consistent with the definition of “technology.”¹⁹³

Judge Smith dissented, holding that the phrase “technological arts ... as used by the examiner is intended to represent a more modern term for the reference to ‘useful arts’ in the Constitution”,¹⁹⁴ which limited the legislative power of Congress and hence the scope of patent law. As such “any laws passed by the Congress to grant patents should be applied in a manner that is consistent with the constitutional mandate”.¹⁹⁵ Judge Smith was critical of the breadth of the majority’s holding noting that:

processes represent an especially troublesome type of invention. This is because almost anything can be claimed as a series of steps that technically can be considered a process, but the term process is so broad that it can be used to claim inventions that cover nothing more than human conduct or thought processes that are totally unrelated to any science or technology.

...

The majority is of the view that every invention is patentable unless it is nothing more than an abstract idea, a law of nature or a natural phenomenon, each of which has been held to be unpatentable by court decisions. These categories of nonstatutory subject matter did not come to us by Scripture however, but instead, resulted from some enlightened individual raising the question of whether such subject matter should be patented. ... The point is that there is no absolute law that says that every category of nonstatutory subject matter has been decided for all time.¹⁹⁶

Judge Smith would have rejected the claims not on a categorical basis as a business method, but on the basis that they were not tied to any science or technology.

Lundgren opened the door in the US to the patenting of non-traditional and clearly non-technical subject matter. One prominent was the patenting of movie scripts, wherein the useful effect is claimed to be “producing valuable entertainment”.¹⁹⁷ Other grants, later derided as being for “the somewhat ridiculous to the truly absurd”¹⁹⁸ covered such non-traditional areas as:

- risk management;
- legal methods;

¹⁹³ *Ex parte Lundgren* Appeal No 2003-2088 (BPAI, 2005) at 87. The reference to “technology” can be understood by reference to Judge Barrett’s earlier discussion at 33 wherein he noted that “a method that does not operate on matter or some form of energy in the physical universe is not ‘useful’ to mankind in the technological sense of the Constitution’s ‘useful arts.’”

¹⁹⁴ *Ex parte Lundgren* Appeal No 2003-2088 (BPAI, 2005) at 10.

¹⁹⁵ *Ex parte Lundgren*. Appeal No 2003-2088 (BPAI, 2005) at 11.

¹⁹⁶ *Ex parte Lundgren*. Appeal No 2003-2088 (BPAI, 2005) at 11-12.

¹⁹⁷ U.S. Patent Application 2005/244804. For the applicant’s legal analysis of the patentability of his ‘invention’, see Andrew F Knight, “A Potentially New IP: Storyline Patents,” (2004) 86(11) *Journal of the Patent and Trademark Office Society* <http://www.plotpatents.com/legal_analysis.htm> (accessed 5 December 2006).

¹⁹⁸ *In re Bilski* 545 F.3d 943 (2008) at 1004. This case is discussed below at 141.

Chapter 3: How Software Became Patentable

- methods of mediation;
- sports moves;
- marketing strategies;
- tax avoidance and estate planning; and
- methods for training janitors, managing toilet reservations and using colour-coded bracelets to designate dating status.¹⁹⁹

As such, it may be suggested that *Lundgren* represents the high water mark of the patentable subject matter inquiry, in that it allowed the scope of patentable subject matter to expand to cover “methods directed to organizing human activity, untethered to technology”.²⁰⁰

By 2006, it appeared that the tide might be beginning to retreat. The Supreme Court, absent from the patentable subject matter debate since *Diehr*, gave some indication that they might be willing to reconsider the boundary lines. For example, in *eBay v MercExchange*,²⁰¹ whilst limiting the availability of permanent injunctive relief for patent infringement suggested that patents for some business methods might be of “suspect validity”.²⁰² Similarly, during oral argument of *Microsoft v AT&T*,²⁰³ Justices Breyer and Scalia suggested a concern at enforcing limits on patentability in the context of the alleged infringement of a software patent.²⁰⁴ However, it would take another 4 years before the Supreme Court finally weighed in, in *Bilski v Kappos*. This case is discussed in detail below.

The United Kingdom

The development of patentable subject matter jurisprudence in the United Kingdom developed over the same period in a largely independent fashion. It may be expected by those viewing the development of modern patent law through a contemporary viewpoint, that because patentable subject matter issues are broadly the same, there would be a great degree of intermingling between these jurisdictions. Perhaps this is merely an Australian perspective, where looking abroad is necessary to make up for the lack of regular litigation of the issue. However, this has not been the case. The differences in constitutional and historical context,²⁰⁵ substantive legal differences,²⁰⁶ the second-tier status of foreign jurisprudence in common law

¹⁹⁹ This list is distilled from Brief for the Respondent, *Bilski v Kappos* 561 US ____ (2010), at 40-42. The last category were given special mention by Breyer J in *Bilski v Kappos* 130 S. Ct. 3218 (2010).

²⁰⁰ Brief for the Respondent, *Bilski v Kappos* 561 US ___, at 42.

²⁰¹ *eBay Inc v MercExchange LLC* 547 U. S. 388 (2006).

²⁰² *eBay Inc v MercExchange LLC* 547 U. S. 388 (2006) at 397 per Kennedy J.

²⁰³ See *Microsoft Corporation v AT&T Corporation (Oral Argument)* 05-1056, 21 February 2007, <http://www.supremecourt.gov/oral_arguments/argument_transcripts/05-1056.pdf> (12 July 2010).

²⁰⁴ *Microsoft Corporation v AT&T Corporation (Oral Argument)* 05-1056, 21 February 2007 at 22 per Breyer J: “I take it that we are operating under the assumption that software is patentable? We have never held that in this Court, have we?”; *ibid* at 13 per Scalia J: “You can’t patent, you know, on-off, on-off code in the abstract, can you? ... There needs to be a device.”

²⁰⁵ See Chapter 2, Section 3 on page 46.

²⁰⁶ UK law remained centred around the concept of manner of manufacture developed since the *Statute of Monopolies*, whilst the US Act set out four categories of patentable subject matter. See Chapter 2, Section 3 on page 46.

legal systems, the lack of interest in harmonisation²⁰⁷ and/or the view of patent law as an instrument of national economic policy may all have contributed to this result. In any event, this independence provides an interesting point of comparison in that it affords an opportunity to see how similar “solutions” have arisen in relation to software patenting and serve as a demonstration of the fact that the problem sought to be solved is not tied to any particular jurisdictional framework.

Developments under the Patents Act 1949 (UK)

In *Slee & Harris’ Application*²⁰⁸ a patent for a linear function solution program was refused. The invention had been claimed as both a method of operating a computer, and a computer when programmed to operate as described. After referring to the Australian case of *NRDC*, and the requirement therein that there be “something in which the new and useful effect be observed”²⁰⁹ the UK Patent Office noted that the product of the “method of operation” claims was “data, i.e. intellectual information”.²¹⁰ As such, it was held that “an end product comprising merely intellectual information is not within the meaning given to the word ‘product’ by the learned Australian Court”.²¹¹ Similarly, it was held that the product of the method, although it may have utility in practical affairs, this was not “essential or inherent in the method claimed”²¹² and as such “the method cannot be regarded as vendible.”²¹³ However, in relation to the second set of claims, the UK Patent Office held that a computer when programmed to operate in the way outlined in the first set of claims amounted to “a machine which is temporarily modified”.²¹⁴ The Patent Office updated its practice notes shortly thereafter to reflect this position.²¹⁵

In *Badger’s Application*,²¹⁶ Lloyd-Jacob J, constituting the Patent Appeal Tribunal considered the patentability of a “method of mechanically designing and forming a visible drawing”,²¹⁷ such

²⁰⁷ Although early steps towards procedural harmonisation were made with the 1883 Paris Convention for the Protection of Industrial Property, it was not until the *Agreement on Trade-Related Aspects of Intellectual Property Rights*, opened for signature 15 April 1994, 1869 UNTS 299, 33 ILM 1197 (entered into force 1 January 1995) (“TRIPS”) came into being that any form of substantive harmonisation came into being at the international level. However TRIPS only prescribes basic requirements for domestic patent laws and fails to provide guidance on substantive requirements. Despite ongoing negotiations within the World Intellectual Property Organisation for substantive patent law harmonisation, no consensus has yet emerged. For an overview of historical developments see John Richards, “Patent Law Harmonization: A Historical Perspective” *Mondaq*, 22 June 2009 <<http://www.mondaq.com/unitedstates/article.asp?articleid=81474>> (13 July 2010); as to the state of ongoing negotiations see World Intellectual Property Organisation “Substantive Patent Law Harmonization” <<http://www.wipo.int/patent-law/en/harmonization.htm>> (13 July 2010).

²⁰⁸ [1966] FSR 51; [1966] RPC 194.

²⁰⁹ *Slee & Harris’ Application* [1966] FSR 51 at 53.

²¹⁰ *Slee & Harris’ Application* [1966] FSR 51 at 53. It should be noted that despite the unfavourable treatment given to Morton’s rules in *NRDC*, the Court in this case continued to refer to the *NRDC* requirements by reference to the concepts of “vendibility” and “product”.

²¹¹ *Slee & Harris’ Application* [1966] FSR 51 at 54.

²¹² *Slee & Harris’ Application* [1966] FSR 51 at 54.

²¹³ *Slee & Harris’ Application* [1966] FSR 51 at 54.

²¹⁴ *Slee & Harris’ Application* [1966] FSR 51 at 55.

²¹⁵ Gary Lea, “The Revolution That Never Was: A Cynic’s Eye View of the Software, Business and E-Commerce Method Patenting Controversy in the Wake of State Street - [2000] DTLJ 4” (2000) 2(1) *Digital Technology Law Journal* 4.

²¹⁶ *Badger Co Inc’s Application* [1969] FSR 474.

²¹⁷ *Badger Co Inc’s Application* [1969] FSR 474 at 475.

Chapter 3: How Software Became Patentable

method being implemented using a “computer and data converter or plotter”.²¹⁸ His Honour found that the method comprised “two distinct aspects, one the preparation, tabulation and codification of the data intended to be fed into the [computer] ... and, second, the conditioning of the operating mechanism so as to permit the computation to proceed in the manner desired”.²¹⁹ His Honour then held that the first aspect was “conceptual in character, lacking in the concrete actuality which differentiates substance from notion, which actuality is not necessarily conferred by writing, printing or otherwise representing the intellectual information upon a sheet, card, tape or other carrier”.²²⁰ As such, “[t]he justification for patent monopoly must then be looked for in the second aspect, namely the conditioning of the competing apparatus so as to control its operation, in consonance with corrections laid down by the data record requirements”.²²¹ Claims drawn in this way were eventually accepted by the Patent Office.²²²

In *Gever’s Application*,²²³ Graham J, considered a claim for a trade mark indexing program. In allowing the appeal against the examiner’s refusal, his Honour found that the claimed invention, was not a “scheme or plan” as characterised by the examiner,²²⁴ but was “data processing apparatus ... constrained to work in a certain way by punched cards which are inserted into it”²²⁵ and therefore a machine, analogous to “a newly invented lathe, the tool of which is constrained by a cam to take a certain path in cutting”²²⁶ or “a new type of carburettor which is accessory to the motor car for which it is intended”.²²⁷ His Honour also held that “it would not be right to class the information or directions which are contained in the punched cards ... as being purely intellectual, literary or artistic [since their object] is amongst other things to ensure that the old machine functions in a particular way”.²²⁸ In that way, it was said that the punched cards “differ from a card which is merely printed and which is intended to convey information to the human eye or mind but which is not intended to be ancillary to some machine being specially shaped or constructed for that purpose.”²²⁹

These early cases indicate a willingness in the UK to find ways in which to allow the patenting of software, in contrast to the reticence displayed by the USPTO. By contrast however, in the same year, the Banks Committee released its Report on the British Patent System²³⁰ which recommended that

a computer programme, that is a set of instructions for controlling the sequence of

²¹⁸ *Badger Co Inc’s Application* [1969] FSR 474 at 476.

²¹⁹ *Badger Co Inc’s Application* [1969] FSR 474 at 476.

²²⁰ *Badger Co Inc’s Application* [1969] FSR 474 at 476.

²²¹ *Badger Co Inc’s Application* [1969] FSR 474 at 476.

²²² *Badger Co Inc’s Application* [1969] FSR 474 at 477-479.

²²³ [1969] FSR 480.

²²⁴ *Gever’s Application* [1969] FSR 480 at 483.

²²⁵ *Gever’s Application* [1969] FSR 480 at 484.

²²⁶ *Gever’s Application* [1969] FSR 480 at 486.

²²⁷ *Gever’s Application* [1969] FSR 480 at 486.

²²⁸ *Gever’s Application* [1969] FSR 480 at 485.

²²⁹ *Gever’s Application* [1969] FSR 480 at 487.

²³⁰ *The British Patent System, Report of the Committee to examine the Patent System and Patent Law* (1970) Cmnd 4407 HMSO. “The terms of reference of the committee included examination of, and reporting on, the British patent system and patent law, in light of the UK Government’s intention to ratify the Strasbourg Convention”: Oliver Mills, *Biotechnological Innovations: moral restraints and patent law* (2005) at 29. The *Strasbourg Convention* 1963 was the precursor to, and model for, the *Convention on the Grant of European Patents*, opened for signature 5 October 1973, 13 ILM 268 (entered into force 7 October 1973), which is discussed further below.

How software became patentable subject matter

operations of a data processing system, in whatever form the invention was presented, should not be patentable. The exclusion would include a method of programming computers, a computer when programmed in a certain way and cases where the novelty or alleged novelty lay only in the programme.²³¹

This position was later supported by the Advisory Group of Non-Governmental Experts on the Protection of Computer Programs held in June 1977 at the World Intellectual Property Organisation.²³²

In *Burroughs Corporation's Application*²³³ the Patent Appeal Tribunal, constituted by Graham and Whitford JJ considered claims for a method of transmitting data over a communication link between computers. In particular, their Honours attempted to reconcile the earlier distinction drawn in *Slee and Harris* with the approach taken in later decisions.

Their Honours rejected the fine distinctions which previous cases had created, such as that between a method of "operating" a computer and a method of "controlling" a computer, since:

the actual words used are not important provided they delimit something more than a mere method or mere idea or mere desideratum, and whether this is so must be judged by looking at all the matters and apparatus in question. The fear that if this claim is allowed it will open the door to methods of operating simple calculators or cover academic mathematical solutions is not, we think, a valid objection. Each case must be looked at on its merits in the light of all the circumstances and a conclusion reached accordingly.²³⁴

In relation to the presently claimed invention, the Tribunal noted that

If the bare method or idea is also clothed by the patentee in his specification with a practical garment in the shape of apparatus enabling that method or idea to be realised in practice, it should no longer be regarded as a naked conception, for it has found a practical embodiment in the apparatus.²³⁵

As such, the present invention was given an appropriate embodiment since "[t]he programme in fact constrains the apparatus to function in a particular way as long as the apparatus embodies that programme".²³⁶ On that basis it could be distinguished from mere method claims such as that in *Rolls Royce's case*²³⁷ where "no actual modification of the mechanism [claimed]

²³¹ James Lahore, "Computers and the Law: The Protection of Intellectual Property" (1978) 5 *Federal Law Review* 15 at 22, citing Banks Committee Report at para 487.

²³² James Lahore, "Computers and the Law: The Protection of Intellectual Property" (1978) 5 *Federal Law Review* 15 at 22.

²³³ [1973] FSR 439.

²³⁴ *Burroughs Corporation's Application* [1973] FSR 439 at 450.

²³⁵ *Burroughs Corporation's Application* [1973] FSR 439 at 449.

²³⁶ *Burroughs Corporation's Application* [1973] FSR 439 at 449.

²³⁷ *Rolls Royce Limited's Application* [1963] RPC 251. The claimed invention in that case, a method for operating and reducing the noise from an aircraft by the pilot was characterised as merely "the disclosure of a general flight plan directing the initial operational movements of an aircraft between take-off and the commencement of free-flying conditions" and therefore "as much outside the operation of any of the useful arts as would be a trainer's direction to a jockey in his control of a racehorse".

Chapter 3: How Software Became Patentable

resulted from the employment of the claimed method”.²³⁸ In *dicta*, the Tribunal noted that on their approach, “computer programmes which have the effect of controlling computers to operate in a particular way, where such programmes are embodied in physical form, are proper subject matter for letters patent.”²³⁹

In the decision of *IBM Corporation’s Application*,²⁴⁰ the Tribunal, again constituted by Graham and Whitford JJ, considered the patentability of a data handling system for a way of determining a selling price “by an automatic computation involving a comparison of buying and selling orders”.²⁴¹ The Tribunal acknowledged that “[s]uch an operation could in theory be done without the need for any automatic aids but in practice needs to be automatically computed.”²⁴² It was also acknowledged that “[i]f claim 1 on its true construction protects [the] business scheme however carried out it must be bad. If it can be so construed as to cover no more than a standard IBM computer it must be bad.”²⁴³ However, these two thoughts were never connected, the Tribunal instead affirming the approach in *Burroughs Corporation*,²⁴⁴ holding that “the claim on its true construction still only covers a computer when it is programmed to produce the required result and does not cover a standard computer as such.”²⁴⁵ The Tribunal had been directed to the US Supreme Court cases existing at that time, but quickly distinguished the invention from that in *Gottschalk v Benson* by holding that “[i]n the case before us there is no attempt to patent a formula or its equivalent.”²⁴⁶ *Parker v Flook* was even more briskly dismissed on the basis that it was “plain that [the US Supreme Court] was not considering the United Kingdom Act of 1949.”²⁴⁷

The European Patent Convention

The *Patents Act* 1977 (UK) greatly altered patent law landscape in the United Kingdom, by bringing the law in line with the European Patent Convention (“EPC”).²⁴⁸ The EPC came into being as a multilateral treaty,²⁴⁹ with a view to removing the administrative and legal difficulties involved in obtaining patent protection in Europe, which up until its adoption required the filing and examination of an application in each country in which the patent was sought.

²³⁸ *Burroughs Corporation’s Application* [1973] FSR 439 at 449.

²³⁹ *Burroughs Corporation’s Application* [1973] FSR 439 at 450.

²⁴⁰ *International Business Machines Corporation’s Application* [1980] FSR 564.

²⁴¹ *International Business Machines Corporation’s Application* [1980] FSR 564 at 567.

²⁴² *International Business Machines Corporation’s Application* [1980] FSR 564 at 567.

²⁴³ *International Business Machines Corporation’s Application* [1980] FSR 564 at 568.

²⁴⁴ *International Business Machines Corporation’s Application* [1980] FSR 564 at 572.

²⁴⁵ *International Business Machines Corporation’s Application* [1980] FSR 564 at 568.

²⁴⁶ *International Business Machines Corporation’s Application* [1980] FSR 564 at 572. This is but one instance of the misguided attempt to draw distinctions between mathematical and non-mathematical subject matter on the basis of the illusory distinction between the numerical and non-numerical algorithms outlined by Newell above.

²⁴⁷ *International Business Machines Corporation’s Application* [1980] FSR 564 at 573.

²⁴⁸ *Convention on the Grant of European Patents*, opened for signature 5 October 1973, 13 ILM 268 (entered into force 7 October 1973). It should be noted that the UK Act “pointlessly uses somewhat different wording although no-one suggests that it has any different meaning.”: *Aerotel Ltd. v. Telco Holdings Ltd. and in the Matter of Macrossan’s Application* [2006] EWCA Civ 1371 (discussed below). On the effect of the EPC on UK law, see for example *Gale’s Application* [1991] RPC 305 at 321-324 per Nicholls LJ (discussed below).

²⁴⁹ It should be noted that the EPC exists independently of the European Union, and its membership is different. A current list of members is available on the EPO website.

How software became patentable subject matter

As such, the Convention harmonised the substantive law between member countries, and also instituted a uniform application procedure administered by a European Patent Office ("EPO").

Whilst it may have resulted in a more efficient administrative procedure for the grant of patents, the EPC has not removed all difficulties. Firstly, the decisions of the EPO and national courts are not binding on each other, leading to a plurality of interpretations of the Convention, not all of which are consistent with each other. Secondly, the EPC did not supplant national patent systems, which continue to operate in parallel, albeit under the substantive law of the EPC. Thirdly, once a patent is granted, there is no single avenue for pursuing infringement proceedings across member states. The issues remain a significant obstacle to complete harmonisation, and are far from resolved.²⁵⁰

Putting aside such problems and turning to the scope of patentable subject matter under the EPC, the starting point is Article 52(2) which declares that the following are not patentable inventions "as such":²⁵¹

- (a) discoveries, scientific theories and mathematical methods;
- (b) aesthetic creations;
- (c) schemes, rules and methods for performing mental acts, playing games or doing business, and programs for computers;
- (d) presentations of information.

Article 53 further outlines a series of policy-based exceptions:

- (a) inventions the publication or exploitation of which would be contrary to "ordre public" or morality, provided that the exploitation shall not be deemed to be so contrary merely because it is prohibited by law or regulation in some or all of the Contracting States;
- (b) plant or animal varieties or essentially biological processes for the production of plants or animals; this provision does not apply to microbiological processes or the products thereof.

Given the express prohibition on the patentability of computer programs, one might expect that software would remain non-patentable in the EU to this day. However, as the history of the non-literal interpretation of "manner of manufacture" outlined in the last chapter should illustrate, such matters are never as so straightforward. As Sherman points out, "specific linguistic analyses ... [are] inevitably dependent upon the current perception of technology and

²⁵⁰ The solution may be the integration of the EPO and creation of a new patent court within the framework of the EU. Work towards such a scheme was commenced in the 1970s, but has repeatedly stalled, and shows no signs of coming into fruition. See Axel H. Horns, "A Unified European Patent System: The Historical Perspective" *IP:JUR*, 18 February 2010, <<http://www.ipjur.com/blog2/index.php?/archives/138-A-Unified-European-Patent-System-The-Historical-Perspective.html>> (12 July 2010).

²⁵¹ The "as such" proviso is found in Article 52(3): "The provisions of paragraph 2 shall exclude patentability of the subject-matter or activities referred to in that provision only to the extent to which a European patent application or European patent relates to such subject-matter or activities as such."

Chapter 3: How Software Became Patentable

as the technology changes these definitions are often rendered obsolete.”²⁵² Thus it is entirely unsurprising that an “imaginative interpretation”²⁵³ was quickly developed by the EPO,²⁵⁴ starting with the observation that from the non-exhaustive list of non-patentable inventions in Article 52(2) share a common characteristic that they are “clearly non-technical”.²⁵⁵ From there “it is a short inductive leap to conclude .. that the term ‘invention’ relates to inventions of a technical nature.”²⁵⁶

Even within this “refined” test however, there remains uncertainty, namely how “the invention as described in the application should be interpreted for the purposes of [Article 52].”²⁵⁷ Sherman notes that four possible approaches thus arise as to how to determine whether an application falls within the exclusions:²⁵⁸

1. a sort and filter approach, where the decision maker should “separate the excluded and non-excluded elements of the application and focus on only the non-excluded components”;
2. the point of novelty approach, wherein “only the novel or inventive elements of the invention are to be examined”, the approach espoused in *Parker v Flook*.
3. the “kernel theory”, wherein “the court should focus only upon the essence or core of the invention”; and
4. the “whole contents approach”, where the court is required to “examine the claims as a whole”, similar to the approach used in *Diamond v Diehr*.

The first three approaches are similar in that they require the examiner “to evaluate and prioritise elements or parts of the invention”.²⁵⁹ Perhaps more significantly however, they “greatly reduce the scope of protection available for computer-related inventions”.²⁶⁰ This is because any software component of a claimed invention will be taken out of consideration. For example, the original EPO Guidelines of 1978 advocated a “contribution” approach, equivalent to the *Flook* point of novelty approach, stating that examiners “should disregard the form or kind

²⁵² Brad Sherman, “The patentability of computer-related inventions in the United Kingdom and the European Patent Office” (1991) 13(3) *European Intellectual Property Review* 85 at 94.

²⁵³ Brad Sherman, “The patentability of computer-related inventions in the United Kingdom and the European Patent Office” (1991) 13(3) *European Intellectual Property Review* 85 at 93.

²⁵⁴ See T16/83 *Christian Francheries/Traffic Regulation* [1988] EPOR 65; T51/84 *Stockburger/Coded Distinctive Mark* [1986] EPOR 229; T208/84 *Vicom/Computer-Related Invention* [1987] EPOR 74 (discussed in detail below); T26/86 *Koch & Sterzel/X-Ray Apparatus* [1988] EPOR 72; T366/87 *Sternheimer/Harmonic Vibrations* [1989] EPOR 131; T115/85 *IBM/Computer-Related Invention* [1990] EPOR 107.

²⁵⁵ Brad Sherman, “The patentability of computer-related inventions in the United Kingdom and the European Patent Office” (1991) 13(3) *European Intellectual Property Review* 85 at 93.

²⁵⁶ Brad Sherman, “The patentability of computer-related inventions in the United Kingdom and the European Patent Office” (1991) 13(3) *European Intellectual Property Review* 85 at 93.

²⁵⁷ Brad Sherman, “The patentability of computer-related inventions in the United Kingdom and the European Patent Office” (1991) 13(3) *European Intellectual Property Review* 85 at 85.

²⁵⁸ Brad Sherman, “The patentability of computer-related inventions in the United Kingdom and the European Patent Office” (1991) 13(3) *European Intellectual Property Review* 85 at 85.

²⁵⁹ Brad Sherman, “The patentability of computer-related inventions in the United Kingdom and the European Patent Office” (1991) 13(3) *European Intellectual Property Review* 85 at 86.

²⁶⁰ Brad Sherman, “The patentability of computer-related inventions in the United Kingdom and the European Patent Office” (1991) 13(3) *European Intellectual Property Review* 85 at 86.

of claim and concentrate on the content in order to identify the novel contribution which the alleged “invention” claimed makes to the known art. If this contribution does not constitute an invention, there is not patentable subject matter.”²⁶¹ In respect of software, this meant that that “[i]f the contribution to the known art resides solely in a computer program then the subject matter is not patentable in whatever manner it may be presented in the claims.”²⁶²

A Working Group was commissioned by the EPO in 1984 to consider the patentability of computer programs under the EPC. The Working Group affirmed the “technical character” test as the correct interpretation based on the negative definition in Article 52, but nonetheless declined to specifically determine what was meant by technology. To date, the giving of such a definition has been avoided.²⁶³

In determining the scope of patentable subject matter in the UK, it is necessary to begin with EPO jurisprudence on the issue. Despite the independence of UK court and the EPO from each other, the legislative intent in giving effect to the EPC means that harmonisation is considered an important objective in deciding cases under that Act.²⁶⁴ The most important early EPO case is thus set out below.

Vicom

Although merely one of six early cases at the EPO, which are hard to reconcile with each other, *Vicom*²⁶⁵ has been given special treatment, especially in the UK, as the exemplar of the correct approach to patentability under the EPC. In *Vicom*, the Board of Appeal at the European Patent Office (EPO) considered the patentability of a method of digitally processing images on a particular piece of hardware.²⁶⁶ The Examiner rejected the claims as directed to a non-patentable mathematical method, for three reasons. Firstly, it was said that this was “because at least the characterising part of the claim would only add a different mathematical concept and would not define new technical subject-matter in terms of technical features.”²⁶⁷ Further, it was “considered that such claims concerned only a mathematical way of approximation of the transfer function”.²⁶⁸ Finally, since digital filtering was considered to be nothing more than “a calcu-

²⁶¹ European Patent Office, *Guidelines for Examination in the EPO* (1978), Chapter C, Part IV, Section 2.2, text reproduced at <<http://eupat.ffii.org/papri/epo-gl78/index.en.html>> (accessed 3 May 2010).

²⁶² European Patent Office, *Guidelines for Examination in the EPO* (1978), Chapter C, Part IV, Section 2.1, text reproduced at <<http://eupat.ffii.org/papri/epo-gl78/index.en.html>> (accessed 3 May 2010).

²⁶³ Most recently, in *G3/08 Programs for computers* (12 May 2010) which is discussed in detail below, the Enlarged Board of Appeal said “[w]e do not attempt to define the term ‘technical’.”: at [9.2].

²⁶⁴ See *Gale’s Patent Application* [1991] RPC 305 at 323 per Nicholls LJ: “The Act did not merely enact the statutory provisions necessary for the provisions of the Convention regarding European patents to take effect in this country. The Act also had a harmonisation objective. ... Accordingly, when construing and applying section 1(1) and (2) of the Act, the court must have regard to the legislative intention with which those subsections were framed, namely, that they were framed so as to have, as nearly as practicable, the same effect in the United Kingdom as the corresponding provisions in Articles 52(1), (2) and (3) of the European Patent Convention have in the territories in which that Convention applies.” Cited with approval in *Fujitsu Ltd’s Application* [1997] RPC 608. As to the limits to this objective see *Symbian Ltd v Comptroller-General of Patents* [2008] EWCA Civ 1066; [2009] RPC 1 at [36] (set out below).

²⁶⁵ *Vicom/Computer-Related Invention* T208/84 [1987] EPOR 74.

²⁶⁶ *Vicom/Computer-Related Invention* T208/84 [1987] EPOR 74 at 74. It was admitted by the applicants however that the claims could be implemented “by a suitably programmed conventional computer.”: *Vicom/Computer-Related Invention* T208/84 [1987] EPOR 74 at 74.

²⁶⁷ *Vicom/Computer-Related Invention* T208/84 [1987] EPOR 74 at 78.

²⁶⁸ *Vicom/Computer-Related Invention* T208/84 [1987] EPOR 74 at 78.

Chapter 3: How Software Became Patentable

lation carried out on two-dimensional arrays of numbers using certain algorithms”,²⁶⁹ it “had therefore to be considered a mathematical operation”.²⁷⁰

The Board of Appeal agreed that “any processing operation on an electric signal can be described in mathematical terms”.²⁷¹ However, the Board noted that

[a] basic difference between a mathematical method and a technical process can be seen, however, in the fact that a mathematical method is carried out on numbers (whatever these numbers may represent) and provides a result also in numerical form, the mathematical method or algorithm being only an abstract concept prescribing how to operate on the numbers. *No direct technical result is produced by the method as such.* In contrast thereto, if a mathematical method is used in a technical process, that process is carried out on a physical entity (which may be a material object but equally an image stored as an electric signal) by some technical means implementing the method and provides as its result a certain change in that entity. *The technical means might include a computer comprising suitable hardware or an appropriately programmed general purpose computer.*²⁷²

The Board went on to note that “a ‘method for digitally filtering data’ remains an abstract notion not distinguished from a mathematical method *so long as it is not specified what physical entity is represented by the data and forms the subject of a technical process*”.²⁷³ It should be questioned whether a representational link between an abstract concept and a physical entity is sufficient to achieve the purpose for which a physicality requirement exists.

The applicant had claimed, in the alternative, apparatus for carrying out the method earlier claimed. Specific hardware for implementing the method was disclosed, although the applicant admitted “that at least in principle it is possible to implement the method and apparatus ... by a suitably programmed conventional computer”.²⁷⁴ Although the Examination Division had dismissed a computer program implementing the method on the basis that “the normal implementation of the claimed methods by a program run on a known computer could not be regarded as an invention in view of Article 52(2)(c) and (3) EPC.”²⁷⁵ The Board considered that such claims were permissible since

a claim directed to a technical process which process is carried out under the control of a program (be this implemented in hardware or in software), cannot be regarded as relating to a computer program *as such* within the meaning of Article 52(3) EPC, as it is the application of the program for determining the sequence of steps in the process for which in effect protection is sought.²⁷⁶

²⁶⁹ *Vicom/Computer-Related Invention* T208/84 [1987] EPOR 74 at 78.

²⁷⁰ *Vicom/Computer-Related Invention* T208/84 [1987] EPOR 74 at 78.

²⁷¹ *Vicom/Computer-Related Invention* T208/84 [1987] EPOR 74 at 79.

²⁷² *Vicom/Computer-Related Invention* T208/84 [1987] EPOR 74 at 79 (emphasis added).

²⁷³ *Vicom/Computer-Related Invention* T208/84 [1987] EPOR 74 at 79.

²⁷⁴ *Vicom/Computer-Related Invention* T208/84 [1987] EPOR 74 at 80.

²⁷⁵ *Vicom/Computer-Related Invention* T208/84 [1987] EPOR 74 at 76.

²⁷⁶ *Vicom/Computer-Related Invention* T208/84 [1987] EPOR 74 at 80.

How software became patentable subject matter

The Board further convinced itself of its conclusion by holding that distinctions between hardware and software were “inappropriate as it can fairly be said that the choice between these two possibilities is not of an essential nature but is based on technical and economic considerations which bear no relationship to the inventive concept as such.”²⁷⁷ Since hardware was clearly patentable, it then followed that software

should not be excluded from protection by the mere fact that for its implementation modern technical means in the form of a computer program are used. Decisive is what *technical contribution* the invention as defined in the claim when considered as a whole makes to the known art.²⁷⁸

It is for this emphasis on “technical contribution” which *Vicom* is usually recited.

Genentech Inc’s Application

In *Genentech*,²⁷⁹ the UK Court of Appeal were largely concerned with issues of inventive step in the context of a biotechnological invention,²⁸⁰ although their Honours took the opportunity to discuss for the first time the impact of the 1977 Act on patent law in the UK, and in particular the approach to be taken with respect to determining patentable subject matter under s1 (the equivalent of Article 52 EPC). In particular, their Honours based their considerations around the decision of the trial judge Falconer J in *Merrill Lynch*²⁸¹ who had held that:

the wording “only to the extent that” [in Article 52(3)/s1] means that there cannot be a patentable invention in so far as the invention resides in the computer program itself, but if some practical (i.e. technical) effect is achieved by the computer or machine operating according to the instructions contained in the program and such effect is novel and inventive (i.e. not obvious), a claim directed to that practical effect will be patentable, notwithstanding it is defined by that computer program.²⁸²

However, the majority in *Genentech* were of the opinion that

while the decision [ie result] in the *Merrill Lynch* Application was probably correct ... the broad expressions of policy were neither necessary for the decision nor can they be reconciled with the judgments ... in *Hickton’s Patent Syndicate* ... Falconer J placed an undue emphasis upon those words which effectively distorted the general meaning of the section itself.²⁸³

²⁷⁷ *Vicom/Computer-Related Invention* T208/84 [1987] EPOR 74 at 80.

²⁷⁸ *Vicom/Computer-Related Invention* T208/84 [1987] EPOR 74 at 80-81 (emphasis added).

²⁷⁹ *Genentech Inc’s Application (Human Growth Hormone)* [1989] RPC 147.

²⁸⁰ Mustill LJ noted that Article 52 issues were “a very late arrival on the scene. If it had been foreshadowed in the pleadings, or explored at the trial, there can be no doubt that the factual position would have been cleared up, so that a proper basis for argument on [subject matter] would have been laid.”: *Genentech Inc’s Patent (Human Growth Hormone)* [1989] RPC 147 at 270.

²⁸¹ *Merrill Lynch’s Application* [1988] RPC 1. This case was subsequently appealed ([1989] RPC 561), and the appeal decision is discussed below.

²⁸² *Merrill Lynch’s Application* [1988] RPC 1 at 12.

²⁸³ *Genentech Inc’s Patent (Human Growth Hormone)* [1989] RPC 147 at 207 per Purchas LJ.

Chapter 3: How Software Became Patentable

From the discussion in Chapter 2, it may be remembered that in *Hickton*'s case, the Court held that it was improper when looking at a claimed invention to excise the idea upon which the invention (or application of the idea) is based, and require that novelty and inventive step be located in that part. Since the approach suggested by Falconer J involved dissecting the invention into excluded and non-excluded parts, and requiring that the requisite novelty and inventive step be found in the non-excluded part, the analogy was irresistible.²⁸⁴ However, Purchas LJ found that the matter was also consistent with the "plain and ordinary interpretation to be given to the words"²⁸⁵ and also the "opinions expressed by the Technical Board of Appeal in *Vicom*'s case".²⁸⁶ In relation to *Vicom*, their Lordships placed particular emphasis on the following passage:

[E]ven if the idea underlying an invention may be considered to reside in a mathematical method, a claim directed to a technical process in which the method is used does not seek protection for the mathematical method as such.²⁸⁷

Also relied on in support of their favoured approach was the decision of the trial judge, Whitford J,²⁸⁸ who held, unlike Falconer J, that the phrase 'only to the extent that' had the effect of restricting "the ban imposed by section 1(2) in relation to the matter set out in [the enumerated exclusions] to inventions which related to these matters *and did not include a manner of using the matter concerned either in a process or in the creation of an artefact*."²⁸⁹

Putting these together, Dillon LJ characterised the approach to dealing with enumerated exclusions, in terms of the exclusion most relevant to the instant case, namely discoveries, as follows:

²⁸⁴ Cf *Genentech Inc's Patent (Human Growth Hormone)* [1989] RPC 147 at 258-259 per Mustill LJ:

We are concerned here with a new statute which differs in important respects from the former law, not only as regards procedure, but also in the balance which it strikes between the interests of the researcher and of the public. We should not assume that the new Act is just the old English law re-written, or that statements of principle or passing observations on individual questions can now be culled from the reported cases and applied without reserve, however eminent the sources from which they are drawn. This is all the more so given that the source of the Act is a treaty, and more-over a treaty written in three languages of equal status.

Whilst it would be senseless to ignore the obviously English parentage of some expressions used in the English text, it would also be a mistake to assume that the Act, which must be interpreted by other national courts and by the tribunals of the European Patent Office, can best be illuminated by researching the arcana of English patent law, distilled from decades of reported cases by the very special analytical processes of the common law. I think it is safer to concentrate on what the new statute says.

See also at 197 per Purchas LJ: "[F]or the purpose of interpreting the 1977 Act it must be viewed in the context of a departure from much of the authority and usage of previous patent law.", and at 235 per Dillon LJ. Cf Dillon J at 240, where his Honour rejects the reasoning of Falconer J on the basis that it "represent a drastic change from English law as previously understood". His Honour does not mention *Vicom*, or any other EPO case law at all. See also *Gale's Patent Application* [1991] RPC 305 at 323 per Nicholls LJ: "When considering [patentability under the new Act], it is helpful to have in mind the principle of patent law, well established before the Act, that an idea or discovery as such is not patentable"; *Gale's Patent Application* [1991] RPC 305 at 330 per Parker LJ.

²⁸⁵ *Genentech Inc's Patent (Human Growth Hormone)* [1989] RPC 147 at 207.

²⁸⁶ *Genentech Inc's Patent (Human Growth Hormone)* [1989] RPC 147 at 208. The relevant principles from *Vicom* are set out in the judgment of Purchas LJ at 206-207.

²⁸⁷ *Vicom/Computer-Related Invention* T208/84 [1987] EPOR 74 at 79.

²⁸⁸ *Genentech Inc's Patent* [1987] RPC 553.

²⁸⁹ *Genentech Inc's Patent (Human Growth Hormone)* [1989] RPC 147 (emphasis added).

How software became patentable subject matter

In so far as a patent claims as an invention the practical application of a discovery, the patent does not, in my judgment, relate only to the discovery as such, even if the practical application may be obvious once the discovery has been made, even though unachievable in the absence of the discovery.²⁹⁰

Merrill Lynch

Given the criticisms of the Court of Appeal in *Genentech*, it was of little surprise when the Court dealing with the *Merrill Lynch* appeal²⁹¹ did not uphold the reasoning of Falconer J. The case is of particular note however because it contained a much clearer attempt to harmonise the UK law with that stated in *Vicom* as representing the correct law in the UK. Fox LJ (with whom Stoker LJ and Taylor LJ agreed) stated the position thus:

It cannot be permissible to patent an item excluded by section 1(2) under the guise of an article which contains that item – that is to say, in the case of a computer program, the patenting of a conventional computer containing that program. Something further is necessary. The nature of that addition is, I think, to be found in the *Vicom* case where it was stated ‘Decisive is what technical contribution the invention makes to the known art.’ There must, I think, be some technical advance on the prior art in the form of a new result.

... If what is produced in the end is itself an item excluded from patentability by section 1(2), the matter can go no further.²⁹²

Gale’s Application

In *Gale’s Application*,²⁹³ Gale claimed to have invented an improved method of calculating the square root of a number using a computer. The claim was specifically directed to a ROM loaded with the program, perhaps seeking to characterise it as patentable hardware as opposed to non-patentable software. Aldous J in considering an appeal from the decision of the examiner, held that

[t]here is a difference between a claim which relates to a disc containing a program and a ROM with particular circuitry. In the former the disc carries the program and therefore can be considered as in effect a claim relating to the program; where as in the latter the program or method is used as the basis for altering the structure of the ROM which then becomes a dedicated piece of apparatus which can be used to carry out the program or method. ...

[A] ROM is more than a carrier, it is a manufactured article having circuit connections which enables the program to be operated. A claim to a ROM with particular

²⁹⁰ *Genentech Inc’s Patent (Human Growth Hormone)* [1989] RPC 147 at 240.

²⁹¹ *Merrill Lynch’s Application* [1989] RPC 561.

²⁹² *Merrill Lynch’s Application* [1989] RPC 561 at 569.

²⁹³ *Gale’s Patent Application* [1991] RPC 305.

Chapter 3: How Software Became Patentable

circuitry, albeit defined by functional steps, cannot to my mind be said to relate to the program or functional steps as such.²⁹⁴

The Court of Appeal disagreed. Their Lordships held that the ROM was “no more than an established type of artefact in which the instructions are physically embedded. It is merely the vehicle used for carrying them.”²⁹⁵ and hence not relevant to characterising the invention. As such, Nicholls LJ (with whom Brown-Wilkinson VC and Parker LJ agreed) approached the subject matter issue on the basis that “it is convenient and right to strip away, as a confusing irrelevance, the fact that the claim is for ‘hardware’. The claim in the specification is, in substance, a claim to a series of instructions which incorporated Mr Gale’s improved method of calculating square roots.”²⁹⁶

As to the objection that the claimed method amounted to a mathematical algorithm, Nicholls LJ noted that “it has a practical application, in that it enables instructions to be written for conventional computers in a way which will, so it is claimed, expedite one of the calculations frequently made with the aid of a computer”²⁹⁷ and did not relate to a mathematical method *as such*. However, this left the issue of whether “the instructions when written, and without more, are not patentable, because they constitute a computer program. Is there something more?”²⁹⁸ The answer depended on whether the program as claimed made a technical contribution. His Lordship held that it did not:

A computer, including a pocket calculator with a square root function, will be a better computer when programmed with Mr. Gale’s instructions. So it may. But the instructions do not embody a technical process which exists outside the computer. Nor, as I understand the case as presented to us, do the instructions solve a “technical” problem lying within the computer, as happened with patent applications such as IBM Corp./Computer-related invention (Decision T 115/85) [1990] E.P.O.R. 107 and IBM Corp./Data processor network (Decision T06/83), [1990] E.P.O.R. 91. *I confess to having difficulty in identifying clearly the boundary line between what is and what is not a technical problem for this purpose.* That, at least to some extent, may well be no more than a reflection of my lack of expertise in this technical field. But, as I understand it, in the present case Mr. Gale has devised an improvement in programming. What his instructions do, but it is all they do, is to prescribe for the CPU in a conventional computer a different set of calculations from those normally prescribed when the user wants a square root. I do not think that makes a claim to those instructions other than a claim to the instructions as such.²⁹⁹

²⁹⁴ *Gale’s Patent Application* [1991] RPC 305 at 316-317.

²⁹⁵ *Gale’s Patent Application* [1991] RPC 305 at 325 per Nicholls LJ.

²⁹⁶ *Gale’s Patent Application* [1991] RPC 305 at 326.

²⁹⁷ *Gale’s Patent Application* [1991] RPC 305 at 327.

²⁹⁸ *Gale’s Patent Application* [1991] RPC 305 at 327 per Nicholls LJ.

²⁹⁹ *Gale’s Patent Application* [1991] RPC 305 at 327-328 per Nicholls LJ (emphasis added). Similar reasoning was applied at 332 per Parker LJ, and at 323 per Browne-Wilkinson VC.

Nicholls LJ is far from alone in his uncertainty as to what is technical and what is not, as “providing an acceptable definition of technology is a very difficult if not impossible task”.³⁰⁰

The EPO Working Group established in 1984 eschewed a definition in favour of illustration by example,³⁰¹ preferring instead to leave the matter to the Technical Board of Appeal.³⁰² The Board however have yet to define the concept. The current approach is encapsulated by the comment “a technical invention seems to be rather like a camel; it is more easily recognised than described”.³⁰³ Sherman notes that the concept may have its origins in German patent law.³⁰⁴

The problem with a lack of definition is that it encourages divergent approaches. Pila notes that over the period the lack of a definition contributed over the period of 1988-1999 to “a series of cases that are difficult to reconcile, if not always in reasoning then at least in conclusion. ... [The effect] was to produce two contradictory lines of cases suggesting, respectively, that programmed, general-purpose computer systems do, and do not, possess the technical character required to avoid exclusion under Art 52(2).”³⁰⁵

Divergence has also taken place between the approaches of the EPO, and UK case law. This will be discussed further below.

Fujitsu Ltd's Application

In this case Fujitsu sought to patent a method and apparatus (a computer program) for modelling a synthetic crystal structure. The examiner “objected that the claimed matter was no more than a scheme rule or method for performing a mental act or a program for a computer and was therefore excluded”.³⁰⁶ The Hearing Officer at the Patents Office upheld the exclusion on both grounds on the basis that the application was “solely concerned with the information content of what was displayed and any possible technical contribution could only come about

³⁰⁰ Brad Sherman, “The patentability of computer-related inventions in the United Kingdom and the European Patent Office” (1991) 13(3) *European Intellectual Property Review* 85.

³⁰¹ Brad Sherman, “The patentability of computer-related inventions in the United Kingdom and the European Patent Office” (1991) 13(3) *European Intellectual Property Review* 85 at 88, footnote 35.

³⁰² Brad Sherman, “The patentability of computer-related inventions in the United Kingdom and the European Patent Office” (1991) 13(3) *European Intellectual Property Review* 85 at 88.

³⁰³ Brad Sherman, “The patentability of computer-related inventions in the United Kingdom and the European Patent Office” (1991) 13(3) *European Intellectual Property Review* 85 at 88, footnote 33. Also noted in Jean Paul Smets and Harmut Pilch, “Software Patentability with Compensatory Regulation: a Cost Evaluation, (2001) 2(6) *Upgrade* 23 <<http://www.upgrade-cepis.org/issues/2001/6/up2-6Smets.pdf>> (4 May 2010) at 25. See also *Aerotel Ltd. v. Telco Holdings Ltd. and in the Matter of Macrossan's Application* [2006] EWCA Civ 1371; [2007] RPC 7 per Jacob LJ at [24]: “One is tempted to say that an Art52(2) is like an elephant: you know it when you see it, but you can't describe it in words. Actually we do not think that is right – there are likely to be real differences depending on what the right approach is.” For a light-hearted but critical discussion of the concept of technicality, which *inter alia*, posits that technical means “this is too complicated to explain in a few words, and too difficult for mere mortals to understand, so we would rather you didn't worry about it until you are sufficiently well-informed”: David Pearce “Mewsings on the word ‘technical’” *IPKat*, 19 December 2007 <<http://ipkitten.blogspot.com/2007/12/mewsings-on-word-technical.html>> (accessed 11 May 2010).

³⁰⁴ See Brad Sherman, “The patentability of computer-related inventions in the United Kingdom and the European Patent Office” (1991) 13(3) *European Intellectual Property Review* 85 at 93-94, noting the German Federal Supreme Court decision in the Red Dove case.

³⁰⁵ Justine Pila, “Dispute over the Meaning of ‘Invention’ in Article 52(2) EPC - The Patentability of computer-implemented Inventions in Europe,” (2005) 36 *IIC: International Review of Intellectual Property and Competition Law* 173 at 178. See footnotes 30 and 32 therein for specific examples of divergent authorities.

³⁰⁶ *Fujitsu Ltd's Application* [1996] RPC 511 per Laddie J.

Chapter 3: How Software Became Patentable

through what that information represents.”³⁰⁷ The fact that the representation was of a technical artefact was not considered sufficient to confer a technical character on the whole invention, since the process involved was “no more than a purely intellectual process of substitution”.³⁰⁸

On appeal to the Patents Court, counsel for the Comptroller noted the difficulties faced by the Patent Office in applying the “technical advance” test, as

in practice it is often very difficult to determine whether a particular invention does as a matter of fact involve the sort of technical contribution or result alluded to in the cases. That difficulty ... is compounded by ... recent decisions of the E.P.O Technical Board of Appeal in which the question of what constitutes a technical advance or contribution seems to have been rather more flexibly interpreted than has hitherto been the case.³⁰⁹

Laddie J unpicked the technical contribution approach, noting

the suggestion that the mere presence of a technical contribution is the defining ingredient of a patentable invention is difficult to reconcile with sections 1(2)(a) and (d). Virtually all discoveries, scientific theories and mathematical methods and many methods of presenting information make a technical contribution yet they are clearly excluded from protection. There is nothing in the section or Art. 52 which suggests a logical reason why “technical” discoveries, scientific theories or mathematical methods should be excluded from patent protection but “technical” computer programs, if suitably disguised for example in a claim to a method or a computer, should not.³¹⁰

His Honour then reviewed recent EPO and UK case law, concluding that their cumulative effect was that

1. The types of subject matter referred to in section 1(2) are excluded from patentability as a matter of policy. This is so whether the matter is technical or not.
2. The exclusion from patentability is a matter of substance not form. Therefore the exclusion under section 1(2) extends to any form of passive carrier or recording of excluded subject matter. ...
3. *Prima facie* a computer running under the control of one program is a different piece of apparatus from the same computer running under the control of another program. It follows that a claim to a computer when controlled by a program or to a method of carrying out a process by use of a computer so controlled can be the subject of patent protection. However, because the court is concerned with substance not form, it is not enough ... to

³⁰⁷ *Fujitsu Ltd's Application* [1996] RPC 511 per Hearing Officer Haselden.

³⁰⁸ *Fujitsu Ltd's Application* [1996] RPC 511 at 519 per Hearing Officer Haselden.

³⁰⁹ *Fujitsu Ltd's Application* [1996] RPC 511 at 521 per Laddie J.

³¹⁰ *Fujitsu Ltd's Application* [1996] RPC 511 at 524.

How software became patentable subject matter

seek protection by framing it in one of these terms. The court or patent office must direct its attention not to the fact that the program is controlling the computer but to what the computer, so controlled is doing.

4. ... If all that is being done, as a matter of substance, is the performance of one of the activities defined under section 1(2) as unprotectable, then it is still unprotectable.³¹¹

On this approach, the instant “claims are not to programs as such. It follows that, *prima facie*, they avoid the ‘program’ exclusion. ... The real issue, it seems to me, is whether the application also avoids the other exclusions”.³¹² His Honour held that the claims were excluded as a scheme or method of performing a performing a mental act, since they involved “a significant level of abstraction and intellectual generality”³¹³ and were as a result “determined by the skill and assessment of the operator”³¹⁴.

The Court of Appeal arrived at the same conclusion as Laddie J, and supported the notion that “the decision as to what was patentable depend[s] upon substance not form.”³¹⁵ However in doing so they re-affirmed the technical contribution approach. After noting the comments of Fox LJ in *Merrill Lynch* (above), and confirming that the correct approach was to look to the substance, not the form of the claims, Aldous LJ³¹⁶ noted that

it is and has always been a principle of patent law that mere discoveries or ideas are not patentable, but those discoveries and ideas which have a technical aspect or make a technical contribution are. Thus the concept that what is needed to make an excluded thing patentable is a technical contribution is not surprising. That was the basis for the decision of the Board in *Vicom*. It has been accepted by this court and by the EPO and has been applied since 1987. It is a concept at the heart of patent law.³¹⁷

The Court also rejected a narrow interpretation of the mental steps exclusion, in which the exclusion should “only [cover] methods which the human mind carries out”.³¹⁸ According to their Aldous LJ, such a construction should be rejected for three reasons:

First, a decision as to whether an invention is patentable as consisting of a method of performing a mental act as such should be capable of determination without recourse to evidence as to how the human mind works. Second, the narrow interpretation appears to introduce a consideration of novelty which is covered in section 1(1)(a). Third, the words used as “a mental act” suggest any mental act whether done before or not.”³¹⁹

³¹¹ *Fujitsu Ltd's Application* [1996] RPC 511 at 530-531.

³¹² *Fujitsu Ltd's Application* [1996] RPC 511 at 532.

³¹³ *Fujitsu Ltd's Application* [1996] RPC 511 at 532.

³¹⁴ *Fujitsu Ltd's Application* [1996] RPC 511 at 532.

³¹⁵ *Fujitsu Ltd's Application* [1997] RPC 608 at 614 per Aldous LJ.

³¹⁶ With whom Roch LJ and Leggatt LJ agreed.

³¹⁷ *Fujitsu Ltd's Application* [1997] RPC 608 at 614 per Aldous LJ.

³¹⁸ *Fujitsu Ltd's Application* [1997] RPC 608 at 619 per Aldous LJ.

³¹⁹ *Fujitsu Ltd's Application* [1997] RPC 608 at 620 per Aldous LJ.

Chapter 3: How Software Became Patentable

The IBM Twins

Whilst earlier cases had considered the patentability of computer-related inventions used in the context of industrial processes, increasingly liberal approaches at the EPO saw the eventual recognition of claims to software whose only effect was on the internal operation of a computer.

In *IBM/Computer program product*,³²⁰ the Board considered the patentability of a method of more efficiently executing an application on a computer, using a “two-phase commit procedure”. A number of claims in the form of method and systems claims had been found by the examining division to meet the requirements of patentability. However, claims 20 and 21 sought to patent the method as a “computer program product”, but were rejected as the Guidelines stated that “a computer program claimed by itself is not patentable irrespective of its contents”.³²¹ However, the Board found that such claims were permissible in certain circumstances:

In the view of the Board, a computer program claimed by itself is not excluded from patentability if the program, when running on a computer or loaded into a computer, brings about, or is capable of bringing about, *a technical effect which goes beyond the ‘normal’ physical interactions between the program (software) and the computer (hardware) on which it is run.*³²²

According to the Board, such a “further technical effect” will exist where the effect of running the program has “a technical character, or where they cause the software to solve a technical problem”.³²³ In arriving at this conclusion, the Board noted that it was

illogical to grant a patent for both a method and the apparatus adapted for carrying out the same method, but not for the computer program product, which comprises all the features enabling the implementation of the method and which, when loaded in a computer, is indeed able to carry out that method.³²⁴

The Board also took the opportunity to make the point that there was no requirement of newness required to be shown in this further technical effect:

[F]or the purpose determining the extent of the exclusion ... the said “further” technical effect may, in its opinion, be known in the prior art.

Determining the technical contribution an invention achieves with respect to the prior art is therefore more appropriate for the purpose of examining novelty and inventive step than for deciding on possible exclusion.³²⁵

³²⁰ T1173/97 *IBM/Computer program product* [1999] OJ EPO 609.

³²¹ T1173/97 *IBM/Computer program product* [1999] OJ EPO 609 at [IV].

³²² T1173/97 *IBM/Computer program product* [1999] OJ EPO 609 at [13] (emphasis added).

³²³ T1173/97 *IBM/Computer program product* [1999] OJ EPO 609 at [6.4].

³²⁴ T1173/97 *IBM/Computer program product* [1999] OJ EPO 609 at [9.8].

³²⁵ T1173/97 *IBM/Computer program product* [1999] OJ EPO 609 at [8].

How software became patentable subject matter

Subsequently in T935/97 *IBM's Application*,³²⁶ the Board applied identical reasoning³²⁷ in relation to claims for a program which managed the display of information in overlapping operating system windows. The UK Patent Office issued a practice direction shortly thereafter, changing its practice in line with these decisions.³²⁸

The Board Diverges

In the *PBS Partnership/Pension benefits*³²⁹ the EPO Board of Appeal revisited the *Vicom* technical contribution test. The applicant had claimed a method of controlling a pension benefits program realised by way of a computer program. The examiner rejected the application as directed to a method of doing business only, in that it did not display any requisite technical character. The board held that all the features of the main claim were “steps of processing and producing information and having purely administrative, actuarial and/or financial character”³³⁰ and hence did not “go beyond a method of doing business as such”.³³¹ However, an auxiliary claim framed as an apparatus to give effect to the method *was* held to be patentable subject matter. This was because

In the board's view a computer system suitably programmed for use in a particular field, even if that is the field of business and economy, has the character of a concrete apparatus in the sense of a physical entity, man-made for a utilitarian purpose and is thus an invention within the meaning of Article 52(1) EPC.

...

This means that, if a claim is directed to such an entity, the formal category of such a claim does in fact imply physical features of the claimed subject-matter which may qualify as technical features of the invention.³³²

This strange reversal of previous EPO jurisprudence goes largely unexplained, and reduces the computer program exclusion to one of form and not of substance, as any capable patent attorney would be able to draft a computer program claim as a “suitably programmed computer” to give effect to the method embodied in the program. Given the Board's insistence in *Vicom* that it is the substance rather than the form of the claims which must be considered, sweeping reversal was required:

The Board agrees with the appellant that the contribution approach is not appropriate for deciding whether something is an invention within the meaning of Article 52(1) EPC as the board already noted in the [IBM Twins].³³³

³²⁶ T935/97 *IBM's Application* [1999] RPC 861.

³²⁷ Large sections of the text of this latter decision are word-for-word identical with the earlier decision, hence the common reference to these two cases as the “IBM Twins”.

³²⁸ UK Patent Office, “Practice Direction (PO: Claims to Programs for Computers)” [1999] RPC 563.

³²⁹ T931/95 *PBS Partnership/Controlling pension benefits systems* [2002] EPOR 522.

³³⁰ T931/95 *PBS Partnership/Controlling pension benefits systems* [2002] EPOR 522 at 528.

³³¹ T931/95 *PBS Partnership/Controlling pension benefits systems* [2002] EPOR 522 at 528.

³³² T931/95 *PBS Partnership/Controlling pension benefits systems* [2002] EPOR 522 at 530.

³³³ T931/95 *PBS Partnership/Controlling pension benefits systems* [2002] EPOR 522 at 531.

Chapter 3: How Software Became Patentable

The Board went on to further justify this position as follows:

There is no basis in the EPC for distinguishing between “new features” of an invention and features of that invention which are known from the prior art when examining whether the invention concerned may be considered to be an invention within the meaning of Article 52(1) EPC. Thus there is no basis in the EPC for applying this so-called contribution approach for this purpose.³³⁴

Having thus held that the claims were allowable subject matter, the Board went on to hold that the effect of the contribution approach applied by examiners was sufficient to demonstrate that the invention claimed lacked the requisite inventive step.³³⁵

In *Hitachi/Auction method*³³⁶ the Board of Appeal endorsed the rejection of technical contribution in *Pension benefits*.³³⁷ The new approach to patentability was defined thus:

What matters having regard to the concept of “invention” ... is the presence of technical character which may be implied by the physical features of an entity or the nature of an activity, or it may be conferred to a non-technical activity by the use of technical means.³³⁸

This approach has been called the “any hardware approach”³³⁹ in that the requisite technical character will easily be satisfied by specifying in the claims the use of some hardware, such as a computer. The Board went on to acknowledge the broadness of this interpretation would

include activities which are so familiar that their technical character tends to be overlooked, such as the act of writing using pen and paper. Needless to say, however, this does not imply that all methods involving the use of technical means are patentable. They still need to be new, represent a non-obvious technical solution to a technical problem, and be susceptible of industrial application.³⁴⁰

Perhaps it was the difficulties in accurately defining the nature of technical character which caused the EPO to abandon its previous approach. In any event, it is submitted that this represents an abandonment of patentable subject matter as any kind of substantive limit on patent grants, the relevant technical character requirement having been incorporated into the concept of inventive step. And unsurprisingly, the result was the same as in *Pension Benefits*, namely that the claims were rejected for lack of inventive step.³⁴¹

³³⁴ T931/95 *PBS Partnership/Controlling pension benefits systems* [2002] EPOR 522 at 531.

³³⁵ T931/95 *PBS Partnership/Controlling pension benefits systems* [2002] EPOR 522 at 533-534.

³³⁶ T258/03 *Hitachi/Auction Method* [2004] EPOR 55.

³³⁷ T258/03 *Hitachi/Auction Method* [2004] EPOR 55 at [3.3].

³³⁸ T258/03 *Hitachi/Auction Method* [2004] EPOR 55 at [4.5].

³³⁹ See for example *Symbian Ltd v Comptroller-General of Patents* [2008] EWCA Civ 1066; [2009] RPC 1 at [35]; *Aerotel Ltd. v. Telco Holdings Ltd. and in the Matter of Macrossan's Application* [2006] EWCA Civ 1371; [2007] RPC 7 at [26] (both cases are discussed below).

³⁴⁰ T258/03 *Hitachi/Auction Method* [2004] EPOR 55 at [4.6].

³⁴¹ T258/03 *Hitachi/Auction Method* [2004] EPOR 55 at [5]-[12].

How software became patentable subject matter

One problem with this change of position is however apparent from the reasoning in *Pension benefits*, where the court held that “the improvement envisaged by the invention according to the application is an essentially economic one ... which, therefore, cannot contribute to inventive step”.³⁴² It is submitted that this in effect applies the business method exclusion in the inventive step context. This may be permissible, in that subject matter exclusions are undoubtedly related to inventive step (since the inventive step requirement evolved into a separate requirement out of the inherent patentability test). But the business method exclusion is defined in Article 52 as an exclusion on patentable subject matter, so to treat the enumerated exclusions as directly relevant to inventive step *only*, is clearly contrary to the intentions of the drafters of the EPC.

In T424/03 *Microsoft/Clipboard Formats* [2006] EPOR 39 the Board completed its *reductio ad absurdum* with the following statement:

[T]he Board would like to emphasise that a method implemented in a computer system represents a sequence of steps actually performed and achieving an effect, and not a sequence of computer-executable instructions (i.e. a computer program) which just have the potential of achieving such an effect when loaded into, and run on, a computer. Thus, the Board holds that the claim category of a computer-implemented method is distinguished from that of a computer program. Even though a method, in particular a method of operating a computer, may be put into practice with the help of a computer program, a claim relating to such a method does not claim a computer program *in the category of a computer program*.³⁴³

The notion that there is some difference between claims to “a method implemented in a computer system”, which could only be implemented by a computer program, and claims to a computer program which implements such a method is clearly ridiculous, and in all respects untenable. As the emphasised reference in the quote above illustrates, the Board have reduced the exception to a mere form requirement.

The Board also held that the claimed methods contributed to the technical character of the invention in that “[t]he claimed steps ... provide a general purpose computer with a further functionality”³⁴⁴ and a computer has technical character because it is “a technical product involving a character ... Moreover, the computer-executable instructions have the potential of achieving [a] further technical effect of enhancing the internal operation of the computer.”³⁴⁵

Software patent directive

In 2002, the European Commission proposed legislation,³⁴⁶ which became known as the Software Patents Directive which, if passed by the European Parliament would, under the banner

³⁴² T931/95 *PBS Partnership/Controlling pension benefits systems* [2002] EPOR 522 at 532.

³⁴³ T424/03 *Microsoft/Clipboard formats I* [2006] EPOR 39 at [5.1].

³⁴⁴ T424/03 *Microsoft/Clipboard formats I* [2006] EPOR 39 at [5.2].

³⁴⁵ T424/03 *Microsoft/Clipboard formats I* [2006] EPOR 39 at [5.2].

³⁴⁶ *Proposal for a Directive of the European Parliament and of the Council on the Patentability of Computer-Implemented Innovations*, presented by the Commission, COM(2002) 92 final, 2002/0047 (COD), Brussels, 20 February 2002.

Chapter 3: How Software Became Patentable

of harmonisation, have adopted the practice of the EPO in relation to the granting of software patents, in essence adopting the *Pension benefits* approach.³⁴⁷

On September 24, 2003, the European Parliament passed a significantly amended version of the Directive, which reinstated the technical contribution test and strengthened its interpretation against the software patent claims. The amended Directive was next examined by the Council of Ministers, who approved a “compromise document” which in effect reversed the Parliament’s amendments.³⁴⁸ This compromise was rejected by the Parliament on 6 July 2005 by a huge majority of 648 votes to 14, with 18 abstentions.³⁴⁹

Despite the success of the anti-software-patent camp in having the Directive blocked, the EPO case law developments over the same period illustrate that the EPO position has broadened, and the status quo continues in the wake of the Parliament’s ultimate rejection of the Directive. So although the events just described in effect had no effect on the state of the law, they are illustrative in that they demonstrate that nearly 40 years after the US Presidential Commission recommended to Congress that patents not be available for software, the patentability of software and related inventions remains a highly contested issue.

Australia

Early Patent Office practice

The first decision on the patentability of computer programs in Australia, *Philips’ Application*,³⁵⁰ was reported in 1966, the same year a US Presidential Commission recommended against patents for software.³⁵¹ The case concerned a claim for an “electronic computer comprising in combination a directly accessible storage, an indirectly accessible storage and *control circuit permitting block transfers of information from the indirectly accessible storage to the directly accessible storage ...*”.³⁵² Although framed as a claim to a computer, the reference to a control circuit illustrates that this was a claim to software in a technologically primitive form. The ex-

³⁴⁷ Article 4 of the Proposal required that “Member States shall ensure that a computer-implemented invention is patentable on the condition that it is susceptible of industrial application, is new, and involves an inventive step. 2. Member States shall ensure that it is a condition of involving an inventive step that a computer-implemented invention must make a technical contribution.” *Proposal for a Directive of the European Parliament and of the Council on the Patentability of Computer-Implemented Innovations*, presented by the Commission, COM(2002) 92 final, 2002/0047 (COD), Brussels, 20 February 2002. The European Economic and Social Committee, a group of industry representatives, published an Opinion in 2002 which called the Directive “a *de facto* acceptance and justification of the *a posteriori* drift of EPO jurisprudence”: European Economic and Social Committee, “Opinion of the Economic and Social Committee on the ‘Proposal for a Directive of the European Parliament and of the Council on the patentability of computer-implemented inventions’” COM(2002) 92 final — 2002/0047 (COD) (19 September 2002) at [3.1.1].

³⁴⁸ The text of the “compromise” or “common understanding” document is set out in Robert Bray, “The European Union ‘Software Patents’ Directive: What Is It? Why Is It? Where Are We Now?” [2005] *Duke Law and Technology Review* 0011.

³⁴⁹ Lucy Sherriff, “EU Parliament bins software patent bill” *The Register*, 6 July 2005 <http://www.theregister.co.uk/2005/07/06/eu_bins_swpat/> (accessed 17 May 2010).

³⁵⁰ *NV Philips Gloeilampenfabrieken’s Application* (1966) 36 AOJP 2392.

³⁵¹ Discussed above at 79.

³⁵² *NV Philips Gloeilampenfabrieken’s Application* (1966) 36 AOJP 2392 at 2392 (emphasis added).

How software became patentable subject matter

aminer rejected the claims, stating that the programming of a known computer amounted to a scheme for the use of old apparatus, and therefore non-patentable.³⁵³

In 1968, the Australian Patent Office in *British Petroleum's Application*³⁵⁴ rejected the patenting of computer programs as both mischievous to the state and generally inconvenient. The examiner characterised the claimed invention as “a new way of solving a specified group of mathematical problems or with a new way of programming a computer to solve such problems”.³⁵⁵ According to the examiner it was “obvious from the very definition of ‘invention’ that a new way of solving mathematical problems is not patentable, no matter how meritorious or ingenious the new solution may be”³⁵⁶ and this was not changed by claiming it as the solving of such a problem using a computer.³⁵⁷ Further, the examiner noted that even if it was assumed to be a manner of manufacture, the claims were still to be excluded as “generally inconvenient”, since

[c]omputer programming is a relatively young art and, although many stratagems and simplifications have been devised so far, a much greater number may be expected to be devised in the future. It would certainly be mischievous to the State and generally inconvenient if, after inventing a million dollars in a computer, the owner were to find himself prevented from operating it efficiently, or in any other manner he may wish, or with any degree of privacy or secrecy he may desire.

It has accordingly been the practice of the Patent Office to refuse applications for patents where the invention consists merely of a programme or working directions for a known computer.³⁵⁸

That same year, *Texas Instruments Inc's Application*³⁵⁹ considered the patentability of “a method of processing seismic traces obtained by geophysical exploration”.³⁶⁰ The hearing officer characterised the claims as “directed to a method of doing something with the traces of a seismic record, which will result in the indication of the presence of some particularised signals.”³⁶¹ The patentability of the invention was rejected on a number of grounds. Firstly, it was said that although “the claims do not contain any mathematical symbols . . . the enumerated steps clearly represent arithmetic [sic] operations.”³⁶² Further, it was noted that “[t]he claim is not essentially concerned with any particular kind of equipment. It starts with information in the form of a seismic record, and proposes to manipulate that information in such a manner that

³⁵³ In support of that proposition the examiner relied on *British United Shoe Machinery Co. Ltd. v. Simon Collier Ltd* (1909) 26 RPC 21 (discussed in Chapter 2).

³⁵⁴ *British Petroleum Ltd's Application* (1968) 38 AOJP 1020.

³⁵⁵ *British Petroleum Ltd's Application* (1968) 38 AOJP 1020 at 1020.

³⁵⁶ *British Petroleum Ltd's Application* (1968) 38 AOJP 1020 at 1020.

³⁵⁷ *British Petroleum Ltd's Application* (1968) 38 AOJP 1020 at 1020.

³⁵⁸ *British Petroleum Ltd's Application* (1968) 38 AOJP 1020 at 1021, citing in support *Rolls-Royce Ltd's Application*, (1963) RPC 251, at 255 per Lloyd-Jacob LJ.

³⁵⁹ *Texas Instruments Inc's Application* (1968) 38 AOJP 2846.

³⁶⁰ *Texas Instruments Inc's Application* (1968) 38 AOJP 2846 at 2846.

³⁶¹ *Texas Instruments Inc's Application* (1968) 38 AOJP 2846 at 2847.

³⁶² *Texas Instruments Inc's Application* (1968) 38 AOJP 2846 at 2847.

Chapter 3: How Software Became Patentable

further information is derived.”³⁶³ As such, it was “directed to a method of operating upon results obtained from the use of admittedly known apparatus”.³⁶⁴

Finally, the hearing officer directed his attention to the requirements spelled out in *NRDC*. It was submitted on behalf of the applicant that “oil exploration, in which this invention may be useful, clearly relates to the field of economic endeavour”.³⁶⁵ However, the hearing officer found that

a process of mathematical operations performed on a set of curves representing mathematical functions appears to me to lie in the realm of fine arts, in the sense that intellectual rather than industrial activity is involved, and that seems to be the case regardless of whether the operations are carried out mentally, or with the aid of a slide rule, or with any other type of computational assistance.³⁶⁶

Further, the hearing officer noted that, in light of the High Court’s comments in *NRDC* about the need to avoid the “fetters of an exact verbal formula”³⁶⁷, the proper way to determine the patentability of the claims was to ask “whether a method of processing seismic information is a proper subject for letters patent”.³⁶⁸ It was asserted that in the instant case it was not since the effect of the patent would be to prevent others from “deriv[ing] information as to the presence of [a particular signal] by following a rather broadly particularised manipulation of the [source information]”.³⁶⁹

The decision in *Ericsson’s Application*³⁷⁰ clearly evidences that in 1974, nearly 20 years after *NRDC*, the understanding of software as of tangential importance to the hardware still dominated, and that the Patent Office practice of denying the patentability of computer programs remained unchanged, despite the liberalisation in the UK discussed above.

Although [the application] appears to limit the field to problem-oriented programmes, it is clear that the nature of the computer to be used will have an effect, and that the programme will have to be expressed and modified to suit the machine. This is quite apart from the fact that the various languages in which the programme is consecutively expressed must also be chosen to suit the type of hardware to be employed. The piece of paper or cardboard or tape or film on which the programme is written, typed, drawn, punched or otherwise recorded will be considered to be *a record of the programme and not the programme itself*.

A programmer is a person who conceives a programme and expresses it in some form. For the purpose of the present considerations, I am taking programming as

³⁶³ *Texas Instruments Inc’s Application* (1968) 38 AOJP 2846 at 2847.

³⁶⁴ *Texas Instruments Inc’s Application* (1968) 38 AOJP 2846 at 2848.

³⁶⁵ *Texas Instruments Inc’s Application* (1968) 38 AOJP 2846 at 2849.

³⁶⁶ *Texas Instruments Inc’s Application* (1968) 38 AOJP 2846 at 2849.

³⁶⁷ *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 271

³⁶⁸ *Texas Instruments Inc’s Application* (1968) 38 AOJP 2846 at 2849.

³⁶⁹ *Texas Instruments Inc’s Application* (1968) 38 AOJP 2846 at 2849-2850.

³⁷⁰ *In Telefon A/B LM Ericsson’s Application* [1975] FSR 49; 44 AOJP 846.

How software became patentable subject matter

referring to the mental and manual processes of originating a programme. Any subsequent routine operations, performed on the programme, such as coding, translating compiling or loading, may also be included in the general operation of programming; but those actions by themselves, that is without the conception of the programme, do not add up to programming.

On the basis of the above definitions, the practice of the patent office in matters relating to programming of computers may be summarised in the following manner. Computer programmes, consisting of sequences of instructions how a problem may be solved, are not proper subject for letters patent. Methods of programming, consisting of the writing down, in one form or another, of a programme are also not a proper subject for letters patent. *A tangible record of a programme in a physical form may be proper subject-matter for letters patent if it can be differentiated from the prior art by features other than the recorded text of the instructions.* And finally, a computer, programmed by a particular programme, may also be proper subject-matter for letters patent if its hardware is different from the prior art or has been effectively modified by the programme.

...

It would be indeed very inconvenient if a person having purchased, or hired, at great expense a machine capable of performing various functions in any desired sequence as may be needed and directed, were to be prevented from the use of some sequences or instructions.³⁷¹

Despite this early practice, when the Australian Patent Office updated its guidelines as a part of a major rewrite in 1984, the Freeman-Walter-Abele (FWA) test was adopted as the proper test for patentability of computer-related claims.³⁷²

IBM v Commissioner of Patents

In 1991 in *IBM v Commissioner of Patents*,³⁷³ Burchett J was asked to reconsider the refusal of the patent delegate to award a patent for “a method for producing a visual representation of a curve image from a set of control points”.³⁷⁴ The claims were refused by the examiner on the basis of the FWA test in that the claims wholly pre-empted a mathematical algorithm. After reviewing the relevant US case law relevant to the FWA test, and the decision of the delegate, his Honour stated that “the whole of the context rises up to insist that claim 1 is talking about the operation of computers.”³⁷⁵ It is clear that his Honour considered that no pre-emption of

³⁷¹ *In Telefon A/B LM Ericsson's Application* [1975] FSR 49 at 56.

³⁷² “July 1984: A major revision of the Patent Examiner’s Manual occurred, and all parts were rewritten.”: IP Australia, “Patent Examiners Manual: History” 1 September 2009 <<http://www.ipaustralia.gov.au/pdfs/patentsmanual/WebHelp/introduction/history.htm>> (27 May 2010). The new Guidelines were published in March 1986, see Australian Patent Office, ‘Guidelines for Considering the Patentability of Computer Program Related Inventions’, Canberra, March 1986.

³⁷³ (1991) 33 FCR 218.

³⁷⁴ *IBM v Commissioner of Patents* (1991) 33 FCR 218 at 220.

³⁷⁵ *IBM v Commissioner of Patents* (1991) 33 FCR 218 at 223.

Chapter 3: How Software Became Patentable

the algorithm took place, since the mathematical “formula [sic] is applied to achieve an end, the production of the improved curve image. A method of producing that by computer, which is novel and inventive, is entitled to the protection of the patent laws.”³⁷⁶

Despite this affirmation of the US approach, his Honour spent a significant proportion of his judgment analysing the claims in the context of the requirements of *NRDC* and its precursors. As such, the case was subsequently taken to have rejected the use of the FWA test.³⁷⁷

His Honour firstly set out the facts of *NRDC* before pointing out that in respect of a new use of a known substance their Honours had stated that “there may be invention in the suggestion that the substance may be used to serve the new purpose; and then, provided that a practical method of so using it is disclosed and that the process comes within the concept of [manner of manufacture], all the elements of a patentable invention are present.”³⁷⁸

Burchett J then noted the High Court’s remarks about the misleading nature of any distinction between discovery and invention, and their emphasis on the breadth of the concept of manner of manufacture in modern patent law.³⁷⁹ Most significantly it seems (given subsequent interpretation) was his Honour’s treatment of the following statement:

a process, to fall within the limits of patentability ... must be one that offers some advantage which is material, in the sense that the process belongs to a useful art as distinct from a fine art ... that its value to the country is in the field of economic endeavour.³⁸⁰

From this statement his Honour held that “[i]t is by this, *by the production of some useful effect*, that patent law has distinguished, so far as it has distinguished, between the discovery of a principle of science and the making of an invention.”³⁸¹ In support of this proposition, Burchett J relied upon the statement of Buller J in *Boulton v Bull*³⁸² that a patentable invention “must be for some new production from [a principle], not for [the principle] itself”. His Honour was also strongly influenced by the decision in *Burroughs Corporation’s Application*,³⁸³ in particular the passage which characterised the computer program therein as “clearly directed to a method involving the use of apparatus modified or programmed to operate in a new way”,³⁸⁴ finding that such a method satisfied the *NRDC* requirements as it results “in some improved or

³⁷⁶ *IBM v Commissioner of Patents* (1991) 33 FCR 218 at 226. In support of that approach, his Honour relied upon the statement of Rehnquist J in *Diamond v Diehr* (1981) 450 US 175 at 191-193 that “an inquiry must be made as to whether the claim is seeking patent protection for that formula in the abstract. A mathematical formula as such is not accorded the protection of our patent laws. ... Because we do not view the respondents’ claims as an attempt to patent a mathematical formula, but rather to be drawn to an industrial process ... we affirm the judgment of the Court of Customs and Patent Appeals”.

³⁷⁷ For example, the Patent Examiners Manual was revised as a result, removing the FWA test in favour of his Honour’s distillation of the *NRDC* approach. See Patent Examiners Manual 35.65, August 1992.

³⁷⁸ *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 262, cited in *IBM v Commissioner of Patents* (1991) 33 FCR 218 at 224.

³⁷⁹ *IBM v Commissioner of Patents* (1991) 33 FCR 218 at 224.

³⁸⁰ *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 275, as cited in *IBM v Commissioner of Patents* (1991) 33 FCR 218 at 224.

³⁸¹ *IBM v Commissioner of Patents* (1991) 33 FCR 218 at 224 (emphasis added).

³⁸² *Boulton & Watt v Bull* (1795) 126 ER 651, discussed in Chapter 2.

³⁸³ *Burroughs Corporation (Perkin’s) Application* [1973] FSR 439, discussed above.

³⁸⁴ *Burroughs Corporation (Perkin’s) Application* [1973] FSR 439 at 449.

modified apparatus, or an old apparatus operating in a novel way, with consequent economic importance or advantages in the field of the useful as opposed to the fine arts”.³⁸⁵ Following this approach, Burchett J was satisfied that “the production of an improved curve image is a commercially useful effect in computer graphics”³⁸⁶ and hence patentable subject matter.

However, it is submitted that it is not immediately clear how the quoted passage from *NRDC* necessarily resolves to the “the production of a useful effect”, which leans dangerously close to the sort of “exact verbal formula” which *NRDC* itself warned against. Looking for a useful effect neither does not answer the question of whether the instant invention is “a proper subject of letters patent”, or indeed whether claims to the implementation of a mathematical algorithm in software belong in the fine arts as opposed to the useful arts.

The role of analogy also features strongly in the decision, his Honour noting that the *NRDC* court had said that “[i]n the varying applications of which the word ‘manufacture’ is capable analogy has always played a considerable part.”³⁸⁷ The central analogy drawn is between *NRDC*’s invention and IBM’s, it being said that “the use of the algorithm is not conceptually different from the use of the compounds involved in [*NRDC*]”³⁸⁸ and also noting that the process claimed was similar to the use of a mathematical algorithm as a step in an “industrial process for the molding of rubber” in *Diehr*.³⁸⁹ Of course, there is a significant difference. Both “analogous” processes were directly linked to *physical* processes, the importance of such a physicality to the concept of inherent patentability having been noted both in Chapter 2 and in this chapter. If the EU law just described illustrates nothing else, it is that the extent to which computer hardware represents a physical manifestation of a claim to software is far from resolved. Whilst in a sense it is true that the effect of running the computer program may be physically observable in the form of a visible “improved curve image” on a computer monitor, such a broad causative view ignores the important role that abstraction plays in software development, as outlined in Chapter 1.

CCOM Pty Ltd v Jiejing Pty Ltd

In *CCOM*,³⁹⁰ the Full Federal Court addressed the patentability of “the use of computer technology, for example in word processors, for the storage and retrieval of Chinese characters”.³⁹¹ At the trial, Cooper J found against the appellants on the subject matter issue as follows:

The material feature of the claimed combination is . . . the procedures used to organise and process the data. The other integers of programming and computer hardware are merely a conventional means to produce the desired result. Taking the

³⁸⁵ *Burroughs Corporation (Perkin’s) Application* [1973] FSR 439 at 449.

³⁸⁶ *IBM v Commissioner of Patents* (1991) 33 FCR 218 at 226.

³⁸⁷ *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 272, as cited in *IBM v Commissioner of Patents* (1991) 33 FCR 218 at 224.

³⁸⁸ *IBM v Commissioner of Patents* (1991) 33 FCR 218 at 225.

³⁸⁹ *IBM v Commissioner of Patents* (1991) 33 FCR 218 at 226.

³⁹⁰ *CCOM Pty Ltd v Jiejing Pty Ltd* (1994) 51 FCR 260.

³⁹¹ *CCOM Pty Ltd v Jiejing Pty Ltd* (1994) 51 FCR 260 at 265. In particular, the invention was characterised as “a search of [a] database by stroke category and by order, so that retrieval and display of Chinese characters are dependent upon satisfaction of search criteria of character stroke-type categories and the order in which those categories are entered through the input entry keys”: at 271.

Chapter 3: How Software Became Patentable

claim as a whole, that which is sought to be made the subject matter of a monopoly is the use of stroke-type categories and stroke order as defined in the specification to organise and process data relevant to Chinese characters in a database and to retrieve and display Chinese characters on a computer screen. The formulation of such criteria and their use as rules to organise and process data ... are the product of human intellectual activity lying in the fine arts and not the useful arts.³⁹²

On appeal, the claims were held to be patentable. The Court was critical of the trial judge's approach, noting that "he was influenced in the determination of the issue as to 'manner of manufacture' by asking whether what was claimed involved anything new and unconventional in computer use."³⁹³ However, according to their Honours, this was to confuse the "distinct requirements of a patentable invention"³⁹⁴ namely manner of manufacture, novelty, inventiveness and utility. Their Honours noted that the manner of manufacture requirement "after the *NRDC* case ... required a decision as to what properly and currently falls within the scope of the patent system ... [I]n so far as 'manufacture' suggests a 'vendible product', this is to be understood as covering every end produced or artificially created state of affairs which is of utility in practical affairs and whose significance is thus economic."³⁹⁵

As to the relevance of UK law, the Court noted a "deliberate legislative choice"³⁹⁶ not to follow the 1977 UK Act, and that beyond the express exclusion of human beings and biological processes for their generation, the legislature "left the matter, in terms of s 18(1)(a), to rest with the concept of manner of manufacture within the meaning of s 6 of the *Statute of Monopolies*, as developed by the Courts, notably in the *NRDC* case."³⁹⁷ However, there was still "significant guidance to be obtained from the course of decisions in Britain before the new legislation with the application in this field of the principles expounded in the *NRDC* case."³⁹⁸ Their Honours took a similar view of the submissions of counsel for the respondents that "what is involved [in the claims] is the storage of the result of linguistic tasks undertaken 'outside the computer'; what the patentees claim is no more than a desirable characteristic of the computer program",³⁹⁹ relying on *Phillips v Mirabella*⁴⁰⁰ for the proposition that "it is not an invention, or a manner of new manufacture, for someone to specify the criteria required to be met, in the manufacture of a known product from known materials, in order to achieve vendibility."⁴⁰¹ However, the Court held otherwise, asserting that "in a given case objections of this type might found an attack for obviousness or lack of novelty, or for failure to comply with s

³⁹² *CCOM Pty Ltd v Jieijing Pty Ltd* (1993) 27 IPR 577 at 594, cited in *CCOM Pty Ltd v Jieijing Pty Ltd* (1994) 51 FCR 260 at 286.

³⁹³ *CCOM Pty Ltd v Jieijing Pty Ltd* (1994) 51 FCR 260 at 291.

³⁹⁴ *CCOM Pty Ltd v Jieijing Pty Ltd* (1994) 51 FCR 260 at 291.

³⁹⁵ *CCOM Pty Ltd v Jieijing Pty Ltd* (1994) 51 FCR 260 at 291, citing in support *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 276-277.

³⁹⁶ *CCOM Pty Ltd v Jieijing Pty Ltd* (1994) 51 FCR 260 at 289, citing in particular the discussion thereof in *NV Philips Gloeilampenfabrieken v Mirabella International Pty Ltd* (1993) 44 FCR 239 at 267-270.

³⁹⁷ *CCOM Pty Ltd v Jieijing Pty Ltd* (1994) 51 FCR 260 at 289.

³⁹⁸ *CCOM Pty Ltd v Jieijing Pty Ltd* (1994) 51 FCR 260 at 293.

³⁹⁹ *CCOM Pty Ltd v Jieijing Pty Ltd* (1994) 51 FCR 260 at 294.

⁴⁰⁰ *NV Philips Gloeilampenfabrieken v Mirabella International Pty Ltd* (1993) 44 FCR 239 at 264 citing the trial judge Wilcox J. The decision was affirmed on appeal to the High Court. See *NV Philips Gloeilampenfabrieken v Mirabella International Pty Ltd* (1995) 183 CLR 655.

⁴⁰¹ *NV Philips Gloeilampenfabrieken v Mirabella International Pty Ltd* (1995) 183 CLR 655 at

How software became patentable subject matter

40 in one or other of its aspects”⁴⁰² but did not support “an independent ground of objection as to patentability”.⁴⁰³

Thus the decision in *IBM’s Application*⁴⁰⁴ was used in support of the proposition that “a method involving the operation or control of a computer, such that it was programmed in a particular way to operate in accordance with the inventor’s method [amounted to more] than ‘intellectual information’ ... because the method was involved in the program and in the apparatus in physical form”.⁴⁰⁵ This was so despite the fact that “the scheme was not itself novel, and that a completely standard computer could be programmed to perform it.”⁴⁰⁶ Similarly, their Honours noted Burchett J’s reliance on *Burroughs Corporation’s Application* in *IBM v Commissioner of Patents* to establish the proposition that the application of a non-patentable idea, or mathematical formula, “to achieve an end”⁴⁰⁷ may well be “entitled to the protection of patent law”.⁴⁰⁸

The Court’s ultimate conclusion was stated as follows:

The NRDC case ... requires a *mode or manner of achieving an end result which is an artificially created state of affairs of utility in the field of economic endeavour*. In the present case, a relevant field of economic endeavour is the use of word processing to assemble text in Chinese language characters. The end result achieved is the retrieval of graphic representations of desired characters, for assembly of text. The mode or manner of obtaining this, which provides particular utility in achieving the end result, is the storage of data as to Chinese characters analysed by stroke-type categories, for search including “flagging” (and “unflagging”) and selection by reference thereto.⁴⁰⁹

Welcome Real-time SA v Catuity Inc

*Catuity*⁴¹⁰ concerned an action for infringement of a patent for “a process and a device for the operation of smart cards in connection with traders’ loyalty programs”.⁴¹¹ The claims specifically related to a “[m]ethod of processing coded information during a purchase or payment operation by a customer, holder of a card, with a chip, at a traders in which the contents of the memory of the chip card are read and a coupon is or is not printed on the basis of the information arising from the contents of said memory”⁴¹² and including “a specified algorithmic processing [sic]”.⁴¹³ In essence, the nature of the invention was that it allowed for a “dynamic”

⁴⁰² *CCOM Pty Ltd v Jiejing Pty Ltd* (1994) 51 FCR 260 at 294.

⁴⁰³ *CCOM Pty Ltd v Jiejing Pty Ltd* (1994) 51 FCR 260 at 295.

⁴⁰⁴ *International Business Machines Corporation’s Application* [1980] FSR 564, discussed above.

⁴⁰⁵ *CCOM Pty Ltd v Jiejing Pty Ltd* (1994) 51 FCR 260 at 292.

⁴⁰⁶ *CCOM Pty Ltd v Jiejing Pty Ltd* (1994) 51 FCR 260 at 292.

⁴⁰⁷ *CCOM Pty Ltd v Jiejing Pty Ltd* (1994) 51 FCR 260 at 293.

⁴⁰⁸ *CCOM Pty Ltd v Jiejing Pty Ltd* (1994) 51 FCR 260 at 293.

⁴⁰⁹ *CCOM Pty Ltd v Jiejing Pty Ltd* (1994) 51 FCR 260 at 295.

⁴¹⁰ *Welcome Real-Time SA v Catuity Inc and Ors* (2001) 113 FCR 110.

⁴¹¹ *Welcome Real-Time SA v Catuity Inc and Ors* (2001) 113 FCR 110 at 114.

⁴¹² *Welcome Real-Time SA v Catuity Inc and Ors* (2001) 113 FCR 110 at 120. This is the text of claim one, with all other claims being “a variation of the method of claim 1”: at 121.

⁴¹³ *Welcome Real-Time SA v Catuity Inc and Ors* (2001) 113 FCR 110 at 120.

Chapter 3: How Software Became Patentable

allocation of memory on the smart card, as it was used at particular traders, with the result that “chip cards with a small memory capacity [could] be used across thousands of merchants”.⁴¹⁴

The respondents counter-claimed for revocation, one of the bases being that the invention was not a manner of manufacture. In particular, they argued that:

- “working directions and methods of doing things fell outside the concept [of manner of manufacture]. Directions as to how to operate a known article or machine or to carry out a known process so as to produce an old result were not patentable, even though they may be a different and more efficient method of doing things”,⁴¹⁵
- “[a]lso rejected were patents for methods of calculation and theoretical schemes, plans and arrangements”,⁴¹⁶
- “the Patent was no more than a method or system for using well-known integers ... to operate familiar kinds of loyalty and incentive schemes for customers.”⁴¹⁷
- that *IBM* and *CCOM* should be distinguished because “[a]lthough like the present case ... they occurred in an environment of computing, in each case there was a physically observable effect that met the manner of manufacture requirement”,⁴¹⁸ and
- “that the invention fell within the principle stated in [*Microcell*]”⁴¹⁹ namely that “it is not an inventive idea for which a monopoly can be claimed to take a substance which is known ... and make out of it an article for which its known properties make it suitable, although it has not been used to make that article before”.⁴²⁰

After reviewing *NRDC* and *CCOM* in some detail,⁴²¹ Heerey J noted that the US decision of *State Street* was concerned “an issue analogous to the present case”,⁴²² and in particular noting the Rich J’s rejection of a business method exception in that case, in favour of the notion that “business methods should be subject to the same legal requirements for patentability as applied to any other process or method”.⁴²³ His Honour found that despite the

different historical source owing little or nothing to the *Statute of Monopolies* ... the social needs the law has to serve in that country are the same as in ours. In both countries, in similar commercial and technological environments, the law has to strike a balance between, on the one hand, the encouragement of true innovation by the grant of monopoly, and on the other, freedom of competition.⁴²⁴

⁴¹⁴ *Welcome Real-Time SA v Catuity Inc and Ors* (2001) 113 FCR 110 at 110 (headnote).

⁴¹⁵ *Welcome Real-Time SA v Catuity Inc and Ors* (2001) 113 FCR 110 at 133.

⁴¹⁶ *Welcome Real-Time SA v Catuity Inc and Ors* (2001) 113 FCR 110 at 133.

⁴¹⁷ *Welcome Real-Time SA v Catuity Inc and Ors* (2001) 113 FCR 110 at 133.

⁴¹⁸ *Welcome Real-Time SA v Catuity Inc and Ors* (2001) 113 FCR 110 at 133.

⁴¹⁹ *Welcome Real-Time SA v Catuity Inc and Ors* (2001) 113 FCR 110 at 133.

⁴²⁰ *Commissioner of Patents v Microcell Ltd* (1959) 102 CLR 232 at 249.

⁴²¹ See *Welcome Real-Time SA v Catuity Inc and Ors* (2001) 113 FCR 110 at 133-136.

⁴²² *Welcome Real-Time SA v Catuity Inc and Ors* (2001) 113 FCR 110 at 136.

⁴²³ *Welcome Real-Time SA v Catuity Inc and Ors* (2001) 113 FCR 110 at 137.

⁴²⁴ *Welcome Real-Time SA v Catuity Inc and Ors* (2001) 113 FCR 110 at 137-138.

As such, the patentability of the method and device would depend on the application of *NRDC* rather than its characterisation as a business method. In any event, his Honour noted that this was “not a business method, in the sense of a particular method or scheme for carrying on a business. ... Rather, the Patent is for a method and a device, involving components such as smart cards and POS terminals, *in a business*”.⁴²⁵

As to the argument that *CCOM* and *IBM v Commissioner of Patents* could be distinguished by the presence of a physically observable effect, his Honour noted that the argument “turns on an expression not found in *CCOM* itself. Nor does such a concept form part of the Full Court’s reasoning.”⁴²⁶ As such, his Honour did not accept that such a requirement necessarily existed.⁴²⁷ Heerey J’s ambivalence towards such a requirement left the door open for Australian law to follow the *Lundgren*⁴²⁸ approach into highly abstract (or arguably purely abstract) subject matter. His Honour did however note that if such a requirement did exist, “it is to be found in the writing of new information to the behaviour file and the printing of the coupon”.⁴²⁹

In relation to the *Microcell* argument, his Honour was unwilling to characterise the invention as the “mere new use of a known article”⁴³⁰ on the grounds that its properties “(particularly its limited memory space) presented difficulties which were overcome only after much time and effort”.⁴³¹

4 Recent Developments

By 2004-2005, it seemed as if the progression towards the dismantling of the subject matter requirement was almost complete. *State Street*, *Pension Benefits* and *Catuity* all illustrated a willingness by courts and tribunals to relax traditional limitations in order to allow a new class of highly abstract subject matter to come within the reach of patent law.

In fact, the beginnings of a retreat were being sown. The rejection of the Software Patent Directive in the EU, despite having no impact on the legal position in Europe, demonstrated a widespread and vocal discord with the extended reach of patent law. Further, the backlogs at the USPTO were becoming cause for increasing concern.⁴³² This backlog, combined with the

⁴²⁵ *Welcome Real-Time SA v Catuity Inc and Ors* (2001) 113 FCR 110 at 137 (original emphasis).

⁴²⁶ *Welcome Real-Time SA v Catuity Inc and Ors* (2001) 113 FCR 110 at 137.

⁴²⁷ *Welcome Real-Time SA v Catuity Inc and Ors* (2001) 113 FCR 110 at 137. See also *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 270.

⁴²⁸ *Ex parte Lundgren*. Appeal No 2003-2088 (2005), discussed above.

⁴²⁹ *Welcome Real-Time SA v Catuity Inc and Ors* (2001) 113 FCR 110 at 137.

⁴³⁰ *Welcome Real-Time SA v Catuity Inc and Ors* (2001) 113 FCR 110 at 138.

⁴³¹ *Welcome Real-Time SA v Catuity Inc and Ors* (2001) 113 FCR 110 at 138.

⁴³² “[T]he volume and complexity of patent applications continues to outpace current capacity to examine them. The result is a pending – and growing – application backlog of historic proportions. Patent pendency ... now averages more than two years. In more complex art areas, such as data-processing technologies, average pendency now stands at more than 3 years.” United States Patent and Trademark Office, *Performance and Accountability Report for Fiscal Year 2005*, <<http://www.uspto.gov/web/offices/com/annual/2005/2005annualreport.pdf>> (12 July 2010) at 4. In 1989, the number of total applications pending was 222,755, but by 2005 this number had reached 885,002: United States Patent and Trademark Office, “Fiscal Year 2009: Patent Applications Pending Prior to Allowance” <http://www.uspto.gov/web/offices/com/annual/2009/oai_05_wlt_03.html> (12 July 2010). By 2010, “[o]ver 1.2 million non-provisional patent applications [were] pending examination at the USPTO. Of those, more than 700,000 [had] not received even a preliminary examination.”: Dennis Crouch, “Unreasonable Applicant Delay and the USPTO Backlog” *Patently-O*, 9 July 2010, <<http://www.patentlyo.com/patent/2010/07/unreasonable>>

Chapter 3: How Software Became Patentable

grant of patents of dubious quality, were giving rise to criticisms that the patent system in that country was “broken”.⁴³³

In Australia, the broad reach of patent law was beginning to be questioned, not in the least by examiners at the patent office. The Deputy Commissioner of the APO in *Szabo*⁴³⁴ for example, a patent for a “reverse mortgage” was rejected on the basis that the claims did not “embody in any material manner elements of science or technology”.⁴³⁵

As will be seen, the recent trend seems to be, with the notable exception of the EPO, towards setting rather than removing limits on patentable subject matter.

Contrary to the previous section, this section will deal with the Australian jurisdiction first. The reasons for this are twofold. Firstly, the developments should be laid out in chronological order, since in this way the tendency towards greater limitation can be seen. Secondly, because the Australian case law to date has largely tracked US case law, the most recent developments, which have occurred in the US of *Bilski*, may be indicative of the way in which Australian law will continue to develop. The developments at the EPO, although they do not evidence the same concern for limiting patentable subject matter, form a chronological link between the Australian and US developments.

Australia

In the case of *Grant v Commissioner of Patents*,⁴³⁶ the Full Federal Court considered the merits of an innovation patent for “a method for structuring a financial transaction to protect an in-

patent-applicant-delay-and-the-uspto-backlog.html> (12 July 2010).

⁴³³ “Over the past several years, in newspaper articles and at hearings held by the Federal Trade Commission (FTC) and National Academy of Sciences (NAS), industry executives have complained in growing numbers that the patent system is broken.” The FTC and NAS hearings, and the reports produced as a result are discussed in Carl Shapiro, “Patent System Reform: Economic Analysis and Critique” (2004) 19 *Berkeley Technology Law Journal* 1017; Kurt M. Saunders, “Reforming the Patent System: Two Proposals” (2005) 4(1) *The Technology Report* <<http://www4.ncsu.edu/~baumerdl/The%20Technology%20Report/Revised.Reforming.the.Patent.System.pdf>> (13 July 2010). On the “broken” state of the patent system see also James Bessen and Michael J Meurer, *Patent Failure: how judges, bureaucrats, and lawyers put innovators at risk* (2008). See also Adam B Jaffe and Josh Lerner, *Innovation and its discontents: how our broken patent system is endangering innovation and progress and what to do about it* (2004). The issue had also begun to attract the regular attention of the media. See for example Jeff Hecht, “Boy takes swing at US Patents” *NewScientist*, 17 April 2002, <<http://www.newscientist.com/article/dn2178-boy-takes-swing-at-us-patents.html>> (12 July 2010); Ryan Naraine, “When Software Patents Go Bad” *Internetnews.com*, 13 September 2002, <<http://www.internetnews.com/bus-news/article.php/1462741/When-Software-Patents-Go-Bad>> (12 July 2010); Lauren Weinstein, “Stop the Patent Process Madness” *Wired*, 21 October 2002, <<http://www.wired.com/politics/law/news/2002/10/55831>> (12 July 2010); Hal R Varian, “Patent Protection Gone Awry” *New York Times*, 21 October 2004, <<http://www.nytimes.com/2004/10/21/business/21scene.html>> (12 July 2010); Sara Schaeffer, “Patent No. 6,004,596: Peanut Butter and Jelly Sandwich” *Wall Street Journal*, 5 April 2005 <<http://online.wsj.com/article/0,,SB111266108673297874,00.html>> (12 July 2010); Randall Stross, “Why Bill Gates Wants 3,000 New Patents” *New York Times*, 31 July 2005, <<http://www.nytimes.com/2005/07/31/business/yourmoney/31digi.html>> (12 July 2010); Tuan Nguyen, “Red Hat Says U.S. Patent System Is Broken” *Tom’s Guide*, 8 April 2008, <<http://www.tomsguide.com/us/redhat-patent-lawsuit,news-961.html>> (12 July 2010); L Gordon Crovitz, “Patent Gridlock Suppresses Innovation” *Wall Street Journal*, 14 July 2008, <<http://online.wsj.com/article/SB121599469382949593.html>> (12 July 2010).

⁴³⁴ *Innovation Patent No. 2004100848 in the name of Peter Szabo and Associates Pty Ltd* [2005] APO 24.

⁴³⁵ *Innovation Patent No. 2004100848 in the name of Peter Szabo and Associates Pty Ltd* [2005] APO 24 at 61.

⁴³⁶ *Grant v Commissioner of Patents* (2006) 154 FCR 62.

dividual's assets ... from legal liability... [by utilising] a trust, a gift, a loan and a security".⁴³⁷ Broadly speaking, Grant's invention was a new variety of business method.

Grant argued that the "useful effect" of his invention was "an alternative to accepted asset protection methods that enable people to trade."⁴³⁸ It was also argued that no physical device or application of technology was required by the *NRDC* approach. This can be regarded as an open invitation to hold that the computerisation of business methods was in reality just a drafting requirement, and to acknowledge that in substance, pure business methods had become patentable. As a fallback position however, Grant argued that "if the documents constituting the trust, making the gift, detailing the loan or recording the mortgage are produced by computer, that the test would be satisfied".⁴³⁹

The Court (constituted by Heerey, Kiefel and Bennett JJ) started by noting that *NRDC* continued to be the basis for determining patentability, and that permissible subject matter was to be "determined according to the breadth of the concept and principles which the law has developed for the application of s6 [of the *Statute of Monopolies*]. Those principles are applied flexibly, recognising that developments and inventions are 'excitingly unpredictable'".⁴⁴⁰

The Court noted the US case law which excluded "laws of nature, natural phenomena and abstract ideas", and drew on *Diehr*, *State Street*, *Alappat* and *AT&T v Excel Communications*⁴⁴¹ wherein non-patentable subject matter could become patentable through some form of practical application, where it produced a "useful, concrete and tangible result".⁴⁴² As a result, the Court upheld the revocation of Grant's invention, holding that the claimed invention was no more than an abstract, intangible idea. Specifically, the court held that the invention "[did] not produce any artificial state of affairs, in the sense of a concrete, tangible, physical, or observable effect. ... What Mr Grant's method results in is at best an abstract, intangible situation."⁴⁴³

The Court then went on to discuss how previous cases had met this requirement. In *Catuity*, it was noted that although "there was not a physically observable end result in the sense of a tangible product, the invention involved the application of an inventive method where part of the invention was the application and operation of the method in a physical device."⁴⁴⁴ In *NRDC* "an artificial effect was physically created on the land."⁴⁴⁵ In *CCOM* "as in *State Street* and *AT&T*, there was a component that was physically affected or a change in state or information in a part of a machine. These can all be regarded as physical effects."⁴⁴⁶

The Court then went on to consider "a separate issue as to whether, quite apart from what might be called its lack of physicality, the alleged invention lies in a realm of human endeavour

⁴³⁷ *Grant v Commissioner of Patents* (2006) 154 FCR 62 at 64.

⁴³⁸ *Grant v Commissioner of Patents* (2006) 154 FCR 62 at 69.

⁴³⁹ *Grant v Commissioner of Patents* (2006) 154 FCR 62 at 70.

⁴⁴⁰ *Grant v Commissioner of Patents* (2006) 154 FCR 62 at 64-65, citing *NRDC* at 271. The Court also noted support for a flexible interpretation in the US case of *AT&T v Excel Communications* 172 F.3d 1352 (1999) at 1356: "The sea-changes in both law and technology stand as a testament to the ability of the law to adapt to new and innovative concepts, while remaining true to basic principles."

⁴⁴¹ *AT&T v Excel Communications* 172 F.3d 1352 (1999).

⁴⁴² *Grant v Commissioner of Patents* (2006) 154 FCR 62 at 68.

⁴⁴³ *Grant v Commissioner of Patents* (2006) 154 FCR 62 at 70.

⁴⁴⁴ *Grant v Commissioner of Patents* (2006) 154 FCR 62 at 70.

⁴⁴⁵ *Grant v Commissioner of Patents* (2006) 154 FCR 62 at 70.

⁴⁴⁶ *Grant v Commissioner of Patents* (2006) 154 FCR 62 at 70.

Chapter 3: How Software Became Patentable

outside those in which patents may be granted. The fine arts are one such area.”⁴⁴⁷ Their Honours concluded that

[t]he interpretation and application of the law would not be considered as having, in the words of *NRDC*, an industrial or commercial or trading character, although without doubt it is an area of economic importance (as are the fine arts). The practice of the law requires, amongst other things, ingenuity and imagination which may produce new kinds of transactions or litigation arguments which could well warrant the description of discoveries. But they are not inventions.⁴⁴⁸

Despite the focus on *NRDC* as the primary test,⁴⁴⁹ the Court also acknowledged the continued validity of the traditional exceptions to patentability as discussed above:

It has long been accepted that “intellectual information”, a mathematical algorithm, mere working directions and a scheme without effect are not patentable. This claim is to “intellectual information”, working directions and a scheme. It is necessary that there be some “useful product”, some physical phenomenon or effect resulting from the working of a method for it to be property the subject of letters patent.”⁴⁵⁰

Their Honours also dealt with the question of whether the realm of human endeavour in which patents may be granted can be defined positively as that of science and technology. This limitation was originally put forward in the decision of the Delegate in *Szabo*⁴⁵¹ and adopted by Branson J at first instance.⁴⁵² The Court of Appeal noted the requirement in Article 27(1) of TRIPS⁴⁵³ that “patents be available in all fields of technology, provided that they are, inter alia, capable of industrial application”. However, the Court rejected such a requirement, noting

What is or is not to be described as science or technology may present difficult questions now, let alone in a future which is as excitingly unpredictable now as it was in 1623 or 1959, if not more so. We think that to erect a requirement that an alleged invention be within the area of science and technology would be to risk the very kind of rigidity which the High Court warned against.⁴⁵⁴

The Court was also dismissive of Branson J’s interpretation of the requirement in *NRDC* that the invention be of “value to the country in the field of economic endeavour” as requiring an assessment as to whether the “performance of the invention will . . . add to the economic wealth of Australia or otherwise benefit Australian society as a whole”.⁴⁵⁵ Their Honours noted that

⁴⁴⁷ *Grant v Commissioner of Patents* (2006) 154 FCR 62 at 71.

⁴⁴⁸ *Grant v Commissioner of Patents* (2006) 154 FCR 62 at 71.

⁴⁴⁹ “The decision and reasoning of the High Court in [*NRDC*] provides the basis for a determination of what is patentable”: *Grant v Commissioner of Patents* (2006) 154 FCR 62 at 64.

⁴⁵⁰ *Grant v Commissioner of Patents* (2006) 154 FCR 62 at 73.

⁴⁵¹ *Re Innovation Patent No 2004100848 in the name of Peter Szabo and Associates Pty Ltd* [2005] APO 24.

⁴⁵² *Grant v Commissioner of Patents* *Grant v Commissioner of Patents* [2005] FCA 1100 at [15]–[19].

⁴⁵³ *Agreement on Trade-Related Aspects of Intellectual Property Rights*, opened for signature 15 April 1994, 1869 UNTS 299, 33 ILM 1197 (entered into force 1 January 1995), a treaty to which Australia is a party.

⁴⁵⁴ *Grant v Commissioner of Patents* (2006) 154 FCR 62 at 71.

⁴⁵⁵ *Grant v Commissioner of Patents* [2005] FCA 1100 at [21].

[t]he Commissioner did not support that aspect of the reasoning of the primary judge and we do not find it necessary to discuss the requisite economic benefits of the alleged invention. Further, immediate economic utility may not be able to be demonstrated by some inventions yet to be commercially developed but that is not to say they are not patentable.⁴⁵⁶

Their Honours also regarded the invention as non-patentable subject matter on the grounds that it was a new use of known products with known properties for which their known properties make them suitable, which made them non-patentable in line with *NRDC* and *Microcell*.⁴⁵⁷

Branson J had also suggested that the performance of the invention must be in the public interest, through an analysis of the social cost of the invention as compared to any resultant benefit to the public.⁴⁵⁸ However, their Honours dismissed such considerations as irrelevant:

Once a product or process has been patented, its use is subject to the laws of the land, such as ...those concerned with environmental protection, pharmaceutical product approval and occupational health and safety.

Nor is the Court in a position to determine the balance between social cost and public benefit. Parliament has already made that judgment, as its predecessor did in 1623, by rewarding innovation with time-limited monopoly.⁴⁵⁹

United Kingdom & European Union

In the wake of the reformulation of the approach to excluded subject matter at the EPO, an obvious difficulty arose in the UK, in that the Courts were now subject to two divergent lines of authority. Previous Court of Appeal decisions based on the technical contribution approach remained binding. Yet those same decisions acknowledge the importance of applying a uniform interpretation to that of the EPO,⁴⁶⁰ whose decisions are “persuasive, not prescriptive”.⁴⁶¹ That dilemma eventually presented itself for resolution to the Court of Appeal in the case of *Aerotel/Macrossan*.⁴⁶²

Aerotel/Macrossan

In *Aerotel/Macrossan*, the Court considered the patentability of two claimed inventions. The first involved method and system claims for making prepaid telephone calls from any telephone using a special exchange. The second involved an automated method of collecting all the documents required for company incorporation.

⁴⁵⁶ *Grant v Commissioner of Patents* (2006) 154 FCR 62 at 72.

⁴⁵⁷ See *Grant v Commissioner of Patents* (2006) 154 FCR 62 at 72-73.

⁴⁵⁸ *Grant v Commissioner of Patents* [2005] FCA 1100 at [22].

⁴⁵⁹ *Grant v Commissioner of Patents* (2006) 154 FCR 62 at 72.

⁴⁶⁰ See *Gale's Patent Application* [1991] RPC 305 at 323; *Fujitsu Ltd's Application* [1997] RPC 608 (Court of Appeal) per Aldous LJ.

⁴⁶¹ *CFPH LLC's Application* [2005] EWHC 1589 (Pat) at [56] per Prescott QC.

⁴⁶² *Aerotel Ltd. v. Telco Holdings Ltd and in the matter of Macrossan's Application* [2006] EWCA Civ 1371; [2007] RPC 7.

Chapter 3: How Software Became Patentable

The court noted that interpretation of Article 52 was “not an easy task”⁴⁶³ for several reasons:

- (a) [T]here is no evident underlying purpose behind the provisions as a group ... there is nothing to tell you one way or the other whether they should be read widely or narrowly.
- (b) One cannot form an overall approach to the categories. They form a disparate group – no common overarching concept...
- (c) Some categories are given protection by other intellectual property laws...
- (d) [S]ome categories are so abstract they are unnecessary or meaningless. For instance ... how could a scientific theory ever be the subject of a patent claim in the first place?
- (e) There may be overlap between some of the exclusions themselves and between them the overall requirement that an invention be “susceptible of industrial application”.⁴⁶⁴

The Court further noted that although the principles of treaty interpretation allowed recourse to the *travaux préparatoires* to the EPC to aid with their interpretation,

the *travaux* provide no direct assistance to any of the categories we have to consider. ... What does emerge is that the various categories are the result of various compromises and distinct discussions about each of them. So one can at least find confirmation that no overarching principle was intended.⁴⁶⁵

The Court also directed its attention to a disparity of approaches, both at the EPO and in the UK:

It is clear that a whole range of approaches have been adopted over the years both by the EPO and national courts. Often they would lead or would lead to the same result, but the reasoning varies. One is tempted to say that an Art.52(2) exclusion is like an elephant: you know it when you see it, but you can't describe it in words. Actually we do not think that is right – there are likely to be real differences depending on what the right approach is. Billions (euros, pounds or dollars) turn on it.⁴⁶⁶

In particular, the Court analysed the “mutually contradictory”⁴⁶⁷ approaches which have been used at the EPO:

⁴⁶³ *Aerotel Ltd. v. Telco Holdings Ltd and in the matter of Macrossan's Application* [2006] EWCA Civ 1371; [2007] RPC 7 at [9].

⁴⁶⁴ *Aerotel Ltd. v. Telco Holdings Ltd and in the matter of Macrossan's Application* [2006] EWCA Civ 1371; [2007] RPC 7 at [9].

⁴⁶⁵ *Aerotel Ltd. v. Telco Holdings Ltd and in the matter of Macrossan's Application* [2006] EWCA Civ 1371; [2007] RPC 7 at [9].

⁴⁶⁶ *Aerotel Ltd. v. Telco Holdings Ltd and in the matter of Macrossan's Application* [2006] EWCA Civ 1371; [2007] RPC 7 at [24].

⁴⁶⁷ *Aerotel Ltd. v. Telco Holdings Ltd and in the matter of Macrossan's Application* [2006] EWCA Civ 1371; [2007] RPC 7 at [25].

1. The contribution approach: Article 52(2) applies where “the inventive step resides only in the contribution of excluded matter”.⁴⁶⁸
2. The technical effect approach: Article 52(2) does not apply where “whether the invention defined in the claim makes a technical contribution to the known art”.⁴⁶⁹
3. The ‘any hardware’ approach: Article 52(2) does not apply where “the claim involves the use of or is to a piece of physical hardware”, involving three variants, namely
 - a) the *Pension Benefits* approach wherein a method claim to excluded matter will be bad, but a claim to the apparatus by which the method is carried out will not be. However, the apparatus claim will in any event fail to meet the obviousness requirement since “the notional skilled man must be taken to know about the improved, excluded, method”,⁴⁷⁰
 - b) the *Hitachi* approach, wherein a neither the hardware claim, nor the method of using that hardware claim will be excluded, but both will be bad for obviousness for the same reason as in (a); or
 - c) the *Microsoft/Clipboard Formats* approach wherein both method and apparatus claims will be allowable where the claim is to something “concrete”, but the method is not treated as part of the knowledge of the notional skilled man.

The Court rejected the “any hardware” approach, dismissing the treatment of novel excluded matter as part of the prior art in the *Pension benefits* and *Hitachi* variants as “not intellectually honest”⁴⁷¹ and noting that the *Microsoft* approach would let to the patentability of claims to books with new story lines, or devices loaded with new music which “[e]veryone would agree ... must be bad”.⁴⁷² Their Honours further noted “a clear conflict between the variants”,⁴⁷³ and the “mistaken assumption”⁴⁷⁴ of all three variants that “the various categories of Art.52(2) must have something in common”.⁴⁷⁵ Finally, their Honours were critical of the meaning of “computer program” applied by the variants, which interprets it to mean the instructions “as an abstract thing”⁴⁷⁶ as opposed to “instructions on some form of media”.⁴⁷⁷ Taking the former

⁴⁶⁸ *Aerotel Ltd. v. Telco Holdings Ltd and in the matter of Macrossan’s Application* [2006] EWCA Civ 1371; [2007] RPC 7 at [26].

⁴⁶⁹ *Aerotel Ltd. v. Telco Holdings Ltd and in the matter of Macrossan’s Application* [2006] EWCA Civ 1371; [2007] RPC 7 at [26].

⁴⁷⁰ *Aerotel Ltd. v. Telco Holdings Ltd and in the matter of Macrossan’s Application* [2006] EWCA Civ 1371; [2007] RPC 7 at [26].

⁴⁷¹ *Aerotel Ltd. v. Telco Holdings Ltd and in the matter of Macrossan’s Application* [2006] EWCA Civ 1371; [2007] RPC 7 at [27].

⁴⁷² *Aerotel Ltd. v. Telco Holdings Ltd and in the matter of Macrossan’s Application* [2006] EWCA Civ 1371; [2007] RPC 7 at [27].

⁴⁷³ *Aerotel Ltd. v. Telco Holdings Ltd and in the matter of Macrossan’s Application* [2006] EWCA Civ 1371; [2007] RPC 7 at [28].

⁴⁷⁴ *Aerotel Ltd. v. Telco Holdings Ltd and in the matter of Macrossan’s Application* [2006] EWCA Civ 1371; [2007] RPC 7 at [30].

⁴⁷⁵ *Aerotel Ltd. v. Telco Holdings Ltd and in the matter of Macrossan’s Application* [2006] EWCA Civ 1371; [2007] RPC 7 at [30].

⁴⁷⁶ *Aerotel Ltd. v. Telco Holdings Ltd and in the matter of Macrossan’s Application* [2006] EWCA Civ 1371; [2007] RPC 7 at [31].

⁴⁷⁷ *Aerotel Ltd. v. Telco Holdings Ltd and in the matter of Macrossan’s Application* [2006] EWCA Civ 1371; [2007] RPC 7 at [31].

Chapter 3: How Software Became Patentable

view was said to “render the exclusion without any real content. We think the framers of the EPC really meant to exclude computer programs in a practicable and operable form. They meant to exclude real computer programs, not just an abstract series of instructions.”⁴⁷⁸

If the choice between the remaining two approaches were “open for free decision”,⁴⁷⁹ the Court would have chosen the technical contribution approach, since “[p]atents are essentially about information as to what to make or do. If all the patentee has taught new is something about an excluded category, then it makes sense for the exclusion to apply.”⁴⁸⁰ However, the precedents of *Merrill Lynch, Gale* and *Fujitsu* meant that it was bound to perpetuate the technical effect test. In giving effect to that test, their Honours approved of the four step approach proposed by counsel for the Comptroller, holding it to be “a reformulation in a different order of the *Merrill Lynch* test”.⁴⁸¹

1. Properly construe the claim
2. Identify the actual contribution
3. Ask whether it falls solely within the excluded subject matter
4. Check whether the actual or alleged contribution is actually technical in nature.⁴⁸²

The Court then explained each of the steps, noting that “[a]ny test must involve this first step”. It was then observed that the second step was

an exercise in judgment probably involving the problem said to be solved, how the invention works, and what its advantages are. What has the inventor really added to human knowledge perhaps best sums up the exercise. The formulation involves looking at substance not form – which is surely what the legislator intended.⁴⁸³

The third step was said to be “merely an expression of the ‘as such’ qualification of Art.52(3)”.⁴⁸⁴ The final step “may not be necessary because the third step should have covered that”⁴⁸⁵ although it had to be included in order to “follow *Merrill Lynch* as we must”.⁴⁸⁶

⁴⁷⁸ *Aerotel Ltd. v. Telco Holdings Ltd and in the matter of Macrossan’s Application* [2006] EWCA Civ 1371; [2007] RPC 7 at [31].

⁴⁷⁹ *Aerotel Ltd. v. Telco Holdings Ltd and in the matter of Macrossan’s Application* [2006] EWCA Civ 1371; [2007] RPC 7 at [32].

⁴⁸⁰ *Aerotel Ltd. v. Telco Holdings Ltd and in the matter of Macrossan’s Application* [2006] EWCA Civ 1371; [2007] RPC 7 at [32].

⁴⁸¹ *Aerotel Ltd. v. Telco Holdings Ltd and in the matter of Macrossan’s Application* [2006] EWCA Civ 1371; [2007] RPC 7 at [41].

⁴⁸² *Aerotel Ltd. v. Telco Holdings Ltd and in the matter of Macrossan’s Application* [2006] EWCA Civ 1371; [2007] RPC 7 at [40].

⁴⁸³ *Aerotel Ltd. v. Telco Holdings Ltd and in the matter of Macrossan’s Application* [2006] EWCA Civ 1371; [2007] RPC 7 at [43].

⁴⁸⁴ *Aerotel Ltd. v. Telco Holdings Ltd and in the matter of Macrossan’s Application* [2006] EWCA Civ 1371; [2007] RPC 7 at [45].

⁴⁸⁵ *Aerotel Ltd. v. Telco Holdings Ltd and in the matter of Macrossan’s Application* [2006] EWCA Civ 1371; [2007] RPC 7 at [46].

⁴⁸⁶ *Aerotel Ltd. v. Telco Holdings Ltd and in the matter of Macrossan’s Application* [2006] EWCA Civ 1371; [2007] RPC 7 at [46].

In applying the four-step test to the Aerotel claims, the court found that properly construed, both the system and method claims were directed to “a physical device consisting of various components”.⁴⁸⁷ In determining the contribution, their Honours held that “[i]t is true that [the system] could be implemented using conventional computers, but the key to it is a new physical combination of hardware”.⁴⁸⁸ As a result, the claim was “not just a method of doing business but the use of a new apparatus for such a method”,⁴⁸⁹ which, as required by the fourth step “was clearly technical in nature”.⁴⁹⁰

As for the Macrossan claims however, despite the fact that “[t]he method is clearly intended in practice to be carried out by a user accessing an internet site, ... that is not necessarily so”.⁴⁹¹ Properly construed, the contribution of the claims was “an interactive system which will do the job which otherwise would have been done by a solicitor or company formation agent”.⁴⁹² In relation to the third step, the Court rejected the reasoning of the judge in the Court below that “the exclusion is aimed more at the underlying abstraction of business method”,⁴⁹³ holding instead that “[w]hether as an abstract or generalised activity or as a very specific activity, if it is a method of doing business as such it is excluded.”⁴⁹⁴

The court also upheld the decisions of the hearing officer and Mann J that the claims amount to a computer program as such, since the contribution of Macrossan was “to provide a computer program (in practice probably an interactive website) which can be used to carry out the method.”⁴⁹⁵

The Court’s concern about the contradictory approaches at the EPO also lead it to suggest that “surely the time has come for matters to be clarified by an Enlarged Board of Appeal.”⁴⁹⁶ To that end the Court, whilst acknowledging that it had no power to make such a referral, formulated a set of questions “which might be asked of an Enlarged Board in the hope of encouraging such a reference.”⁴⁹⁷ Such a referral was initially dismissed by the President of the EPO, who in a letter to Lord Jacob claimed that “there are insufficient differences between current Board of Appeal decisions dealing with Article 52” and hence an insufficient basis for a referral.⁴⁹⁸

⁴⁸⁷ *Aerotel Ltd. v. Telco Holdings Ltd and in the matter of Macrossan’s Application* [2006] EWCA Civ 1371; [2007] RPC 7 at [51], [54].

⁴⁸⁸ *Aerotel Ltd. v. Telco Holdings Ltd and in the matter of Macrossan’s Application* [2006] EWCA Civ 1371; [2007] RPC 7 at [53].

⁴⁸⁹ *Aerotel Ltd. v. Telco Holdings Ltd and in the matter of Macrossan’s Application* [2006] EWCA Civ 1371; [2007] RPC 7 at [53].

⁴⁹⁰ *Aerotel Ltd. v. Telco Holdings Ltd and in the matter of Macrossan’s Application* [2006] EWCA Civ 1371; [2007] RPC 7 at [53].

⁴⁹¹ *Aerotel Ltd. v. Telco Holdings Ltd and in the matter of Macrossan’s Application* [2006] EWCA Civ 1371; [2007] RPC 7 at [60].

⁴⁹² *Aerotel Ltd. v. Telco Holdings Ltd and in the matter of Macrossan’s Application* [2006] EWCA Civ 1371; [2007] RPC 7 at [63].

⁴⁹³ *Macrossan’s Application* [2006] EWHC 705 (Ch) at [30], cited in *Aerotel Ltd. v. Telco Holdings Ltd and in the matter of Macrossan’s Application* [2006] EWCA Civ 1371; [2007] RPC 7 at [66].

⁴⁹⁴ *Aerotel Ltd. v. Telco Holdings Ltd and in the matter of Macrossan’s Application* [2006] EWCA Civ 1371; [2007] RPC 7 at [68].

⁴⁹⁵ *Aerotel Ltd. v. Telco Holdings Ltd and in the matter of Macrossan’s Application* [2006] EWCA Civ 1371; [2007] RPC 7 at [73].

⁴⁹⁶ *Aerotel Ltd. v. Telco Holdings Ltd and in the matter of Macrossan’s Application* [2006] EWCA Civ 1371; [2007] RPC 7 at [25].

⁴⁹⁷ *Aerotel Ltd. v. Telco Holdings Ltd and in the matter of Macrossan’s Application* [2006] EWCA Civ 1371; [2007] RPC 7 at [25]. The questions posed are set out in the judgment at [76].

⁴⁹⁸ Letter from Professor Alain Pompidou to Lord Justice Jacob, 22 February 2007

Chapter 3: How Software Became Patentable

Symbian

In *Symbian*,⁴⁹⁹ the Court of Appeal considered an application concerning “a method of accessing data in a dynamic link library in a computing device”.⁵⁰⁰ The Comptroller-General argued that such a program was non-patentable on the basis that Article 52 “has the effect of excluding from patentability any program unless it has a novel effect outside the computer”,⁵⁰¹ such as “a program for an improved system for manufacturing a product, or for performing a new or improved function on any machine . . . unless the function was only performed on the computer itself.”⁵⁰² Symbian argued for Article 52 having a “more limited exclusionary effect: it only excludes programs which do not provide a technical solution to a technical problem.”⁵⁰³ From the outset the Court noted the difficulty in determining the breadth of the computer program exclusion, noting that “the issue is inherently problematic, and inevitably will lead to a degree of inconsistency and uncertainty”,⁵⁰⁴ and further acknowledging that “[e]ither of these interpretations could be correct in light of the rather (and perhaps inevitably) imprecise language of Art.52. As a matter of policy and practicality, there are also arguments both ways.”⁵⁰⁵ The arguments canvassed in the judgment were:

- the impact of the TRIPS requirement that inventions be available in “all fields of *technology*” supports the Symbian focus on technicality;⁵⁰⁶
- that it is arbitrary and unfair to discriminate against improvements in the performance of computers as compared to other machines;⁵⁰⁷
- that the Comptroller’s test gives little weight to the “as such” proviso in Article 52(3);⁵⁰⁸
- Symbian’s test reflects the view of the EPO, although it was acknowledged that the decisions of the Board were not binding on the Court;⁵⁰⁹
- the nature of computer programs means that it is difficult to effectively search for prior art, although this difficulty cannot be avoided entirely in any event;⁵¹⁰
- Symbian’s test is “rather imprecise and arbitrary”, but so too is the effect of the Comptroller’s test. This simply reflects the “inherent uncertainty as to what precisely is excluded by the words ‘computer programs . . . as such’”,⁵¹¹

<<http://www.ipo.gov.uk/p-pn-subjectmatter-letter.htm>> (25 May 2010).

⁴⁹⁹ *Symbian Ltd v Comptroller-General of Patents* [2008] EWCA Civ 1066; [2009] RPC 1 at [2].

⁵⁰⁰ *Symbian Ltd v Comptroller-General of Patents* [2008] EWCA Civ 1066; [2009] RPC 1 at [2].

⁵⁰¹ *Symbian Ltd v Comptroller-General of Patents* [2008] EWCA Civ 1066; [2009] RPC 1 at [19].

⁵⁰² *Symbian Ltd v Comptroller-General of Patents* [2008] EWCA Civ 1066; [2009] RPC 1 at [19].

⁵⁰³ *Symbian Ltd v Comptroller-General of Patents* [2008] EWCA Civ 1066; [2009] RPC 1 at [19].

⁵⁰⁴ *Symbian Ltd v Comptroller-General of Patents* [2008] EWCA Civ 1066; [2009] RPC 1 at [18].

⁵⁰⁵ *Symbian Ltd v Comptroller-General of Patents* [2008] EWCA Civ 1066; [2009] RPC 1 at [20].

⁵⁰⁶ *Symbian Ltd v Comptroller-General of Patents* [2008] EWCA Civ 1066; [2009] RPC 1 at [21].

⁵⁰⁷ *Symbian Ltd v Comptroller-General of Patents* [2008] EWCA Civ 1066; [2009] RPC 1 at [22].

⁵⁰⁸ *Symbian Ltd v Comptroller-General of Patents* [2008] EWCA Civ 1066; [2009] RPC 1 at [23].

⁵⁰⁹ *Symbian Ltd v Comptroller-General of Patents* [2008] EWCA Civ 1066; [2009] RPC 1 at [24].

⁵¹⁰ *Symbian Ltd v Comptroller-General of Patents* [2008] EWCA Civ 1066; [2009] RPC 1 at [25].

⁵¹¹ *Symbian Ltd v Comptroller-General of Patents* [2008] EWCA Civ 1066; [2009] RPC 1 at [26].

- Symbian’s test “gives the computer program a very limited effect” although “the exclusion is far from deprived of any meaning”;⁵¹²
- the Comptroller’s test has been adopted by UKIPO and reflects the usual view of hearing officers;⁵¹³

In particular, the Court noted a concern that Article 52 makes “no reference to any ‘technical’ requirement”⁵¹⁴ and “could easily mean different things to different people”.⁵¹⁵ In the view of the Court, this did not make it “unhelpful or inappropriate”⁵¹⁶ but underscored the importance of tribunals giving clear guidance as to its meaning, “otherwise it has all the disadvantages of the original obscure wording, with the added disadvantage of not even providing the actual legislative test.”⁵¹⁷ To that end their Honours noted the extra-curial remarks of Mellulis J of the Bundesgerichtshof⁵¹⁸ who

deprecated the use of the word “technical”, not least because “when assessing software as such, the program’s interdependence with the technical device makes the technical content hard to deny”. To the same effect in *CFPH LLC’s Application* [2005] EWHC 1589 ... para 14, it was suggested that the use of the description “technical” was “a useful servant but a dangerous master”.⁵¹⁹

Overall, the lack of overwhelming arguments either way, and “in light of the difficulty and importance of the issue”,⁵²⁰ the Court felt it “should seek to steer a relatively unadventurous and uncontroversial course, and we should be particularly concerned to minimise complexity and uncertainty. These aims are not necessarily mutually consistent, but, on this occasion, we believe they are achievable.”⁵²¹ As such, the Court noted that the correct approach to determining the ambit of the exclusion was as follows:

If the Court of Appeal cases give tolerably clear guidance which would resolve the issue on this appeal, then we should follow that guidance, unless it is inconsistent with clear guidance from the Board, in which case we should follow the latter guidance unless satisfied that it is wrong.⁵²²

As such, the Court noted that there were some “decisions of the Board subsequent to *Aerotel* which appear to support the approach disapproved in *Aerotel*”.⁵²³ However, their Honours gave five reasons for declining to depart from *Aerotel*:

⁵¹² *Symbian Ltd v Comptroller-General of Patents* [2008] EWCA Civ 1066; [2009] RPC 1 at [27].

⁵¹³ *Symbian Ltd v Comptroller-General of Patents* [2008] EWCA Civ 1066; [2009] RPC 1 at [28].

⁵¹⁴ *Symbian Ltd v Comptroller-General of Patents* [2008] EWCA Civ 1066; [2009] RPC 1 at [29].

⁵¹⁵ *Symbian Ltd v Comptroller-General of Patents* [2008] EWCA Civ 1066; [2009] RPC 1 at [30].

⁵¹⁶ *Symbian Ltd v Comptroller-General of Patents* [2008] EWCA Civ 1066; [2009] RPC 1 at [30].

⁵¹⁷ *Symbian Ltd v Comptroller-General of Patents* [2008] EWCA Civ 1066; [2009] RPC 1 at [30].

⁵¹⁸ The Bundesgerichtshof, or German Federal Supreme Court, is the highest court for civil matters including patent law in Germany.

⁵¹⁹ *Symbian Ltd v Comptroller-General of Patents* [2008] EWCA Civ 1066; [2009] RPC 1 at [31].

⁵²⁰ *Symbian Ltd v Comptroller-General of Patents* [2008] EWCA Civ 1066; [2009] RPC 1 at [32].

⁵²¹ *Symbian Ltd v Comptroller-General of Patents* [2008] EWCA Civ 1066; [2009] RPC 1 at [51].

⁵²² *Symbian Ltd v Comptroller-General of Patents* [2008] EWCA Civ 1066; [2009] RPC 1 at [36].

⁵²³ *Symbian Ltd v Comptroller-General of Patents* [2008] EWCA Civ 1066; [2009] RPC 1 at [46].

Chapter 3: How Software Became Patentable

Firstly, there is no decision of the Enlarged Board. Not only does that mean that the view of the Board is not as authoritative as it could be; it also suggests that the Board does not consider that the time has arrived for the point to be conclusively determined. Secondly, the approaches in the four decisions are not identical: in particular, one of them appears more consistent with the view preferred in *Aerotel*. Thirdly, we are concerned that, particularly if the passage quoted from *FUJITSU LTD/File search method*⁵²⁴ represents the Board's view, the computer program exclusion may have lost all meaning. Fourthly, it is not as if the English courts are alone in their concern about the approach of the Board, as the observations from the German judiciary [cited above and in *Aerotel*] demonstrate. Fifthly, if this court is seen to depart too readily from its previous, carefully considered approach, it would risk throwing the law into disarray.⁵²⁵

As such, the appeal was dismissed, it being held that the *Aerotel* approach supported the patentability of the invention. The technical contribution was "[a] program which makes a computer operate on other programs faster than prior art operating programs enabled it to do by virtue of the claimed features".⁵²⁶ This contribution was not directed to solely excluded subject matter since "it has the known-on effect of the computer working better as a matter of practical reality"⁵²⁷ and such a contribution was technical "on any view as to the meaning of the word".⁵²⁸

Referral to the Enlarged Board (G3/08)

Days after the decision in *Symbian* was handed down, the new President of the EPO, Alison Brimelow made the referral to the Enlarged Board of Appeal which her predecessor had refused.⁵²⁹ Guidance was sought in light of diverging decisions of the EPO which "have created uncertainty".⁵³⁰

Specifically, the questions address four different aspects of patentability in this field. The first question relates to the relevance of the category of the claim. The other three questions ask where the line should be drawn between those aspects excluded from patentability and those contributing to the technical character of the claimed subject-matter: the second question concerns the claim as a whole; the third, individual features of the claim; and the fourth – relevant for defining the skills of the

⁵²⁴ "The claimed method requires the use of a computer. It is therefore technical in character and constitutes an invention within the meaning of Art.52 ...": T1531/04 *Fujitsu Ltd/File search method* (18 April 2007), cited *Symbian Ltd v Comptroller-General of Patents* [2008] EWCA Civ 1066; [2009] RPC 1 at [45].

⁵²⁵ *Symbian Ltd v Comptroller-General of Patents* [2008] EWCA Civ 1066; [2009] RPC 1 at [46].

⁵²⁶ *Symbian Ltd v Comptroller-General of Patents* [2008] EWCA Civ 1066; [2009] RPC 1 at [59].

⁵²⁷ *Symbian Ltd v Comptroller-General of Patents* [2008] EWCA Civ 1066; [2009] RPC 1 at [59].

⁵²⁸ *Symbian Ltd v Comptroller-General of Patents* [2008] EWCA Civ 1066; [2009] RPC 1 at [59].

⁵²⁹ Letter from Alison Brimelow to Peter Messerli, Chairman of the Enlarged Board of Appeal, 22 October 2008, in *G3/08 Programs for computers* (12 May 2010)

<[http://documents.epo.org/projects/babylon/eponet.nsf/0/dc6171f182d8b65ac125772100426656/\\$file/g3_08_en.pdf](http://documents.epo.org/projects/babylon/eponet.nsf/0/dc6171f182d8b65ac125772100426656/$file/g3_08_en.pdf)> (25 May 2010).

⁵³⁰ European Patent Office, "Patentability of programs for computers" Press Release, 24 October 2008, <<http://www.epo.org/topics/news/2008/20081024.html>> (25 May 2010).

(technically) skilled person – concerns the activity (the programming) underlying the resulting product (the computer program).⁵³¹

The Enlarged Board invited the public to submit written statements on the referral, and around 100 were received.⁵³² On 12 May 2010, the Enlarged Board of Appeal of the European Patent Office handed down its decision, finding that the referral was inadmissible. In order for a referral to be admissible, it must pass the requirements set out in Article 112(1)(b) EPC, namely that they (i) need to be answered to ensure uniform application of the law or are of fundamental importance; and (ii) two Boards have given different decisions on the questions referred. The first requirement was held to be established on the basis of “the economic significance of [computer-implemented inventions] in many technical fields, plus the consequent heated public debate on their patentability and the many cases before [the Boards] and various national courts”.⁵³³ However, the second requirement, that there be “different decisions” of two Boards was not made out. What it required was that there be “a divergence, or better a conflict in the case law making it difficult if not impossible for the Office to bring its patent granting practice into line with the case law of the Boards of Appeal.”⁵³⁴ The Enlarged Board noted that:

[d]evelopment of the law is an essential aspect of its application ... Consequently, legal development as such cannot on its own form the basis for a referral, only because case law in new legal territory does not always develop in linear fashion, and earlier approaches may be abandoned or modified.⁵³⁵

On this basis, the Enlarged Board held that:

Even a radical shift in jurisprudence need not necessarily be construed as a different decision within the meaning of [Article 112] provided that the Board corrects itself and – mostly in explicit fashion – declares its earlier practice to be no longer relevant.⁵³⁶

On this approach it is difficult to see how a referral could, except in the most extreme circumstances, make it over such a hurdle. In relation to the question of the computer program exclusion, the Enlarged Board stated that the only divergence evident from the referral was between T424/03 (Microsoft/Clipboard Formats) and T1173/97 (one of the IBM Twins),⁵³⁷ however “this was a legitimate development of the case law and since T1173/97 has not been followed by any Board on this particular point there is no divergence which would make the referral of this point ... admissible.”⁵³⁸

Thus the Board pointed out that:

⁵³¹ European Patent Office, “Patentability of programs for computers” Press Release, 24 October 2008, <<http://www.epo.org/topics/news/2008/20081024.html>> (25 May 2010).

⁵³² G3/08 *Programs for computers* (Opinion of 12 May 2010) at [II.1].

⁵³³ G3/08 *Programs for computers* (Opinion of 12 May 2010) at [4.2].

⁵³⁴ G3/08 *Programs for computers* (Opinion of 12 May 2010) at [7.2.7].

⁵³⁵ G3/08 *Programs for computers* (Opinion of 12 May 2010) at [7.3.1].

⁵³⁶ G3/08 *Programs for computers* (Opinion of 12 May 2010) at [7.3.4].

⁵³⁷ See G3/08 *Programs for computers* (Opinion of 12 May 2010) at [10.2].

⁵³⁸ G3/08 *Programs for computers* (Opinion of 12 May 2010) at [10.12].

Chapter 3: How Software Became Patentable

[t]he present position of the case law is [that] ... a claim in the area of computer programs can avoid exclusion under Articles 52(2)(c) and (3) EPC merely by explicitly mentioning the use of a computer or a computer-readable storage medium. But ... it is also quite clear from the case law ... that ... a claim which specifies no more than “Program X on a computer-readable storage medium” or “A method of operating a computer according to program X” will always still fail to be patentable for lack of an inventive step under Articles 52(1) and 56 EPC. Merely the EPC article applied is different. While the Enlarged Board is aware that this rejection for lack of an inventive step rather than exclusion under Article 52(2) EPC is in some way distasteful to many people, it is the approach which has been consistently developed since T1173/97 ...⁵³⁹

The Board did acknowledge the difficult position in which this left UK courts, but suggested that the solution was a legislative one:

Even the essentially commendable desire for harmonisation expressed by Lord Justice Jacob in the *Aerotel/Macrossan* judgment can be taken up by the Enlarged Board only to the extent possible under the EPC, even if his suggestion might significantly advance the cause of legal uniformity in Europe. When judiciary-driven legal development meets its limits, it is time for the legislator to take over.⁵⁴⁰

In the wake of this ruling, very little has changed in the UK, with the likely result that the divergence between the EPO and the UK will continue. The five reasons stated by the Court in *Symbian* for declining to depart from *Aerotel*, continue to be determinative.⁵⁴¹ As the comments of the Board only relate to the admissibility of the referral, they provide no authoritative guidance as to the subject matter issue. As such, the UKPO, in the July 2010 update to the Manual of Patent Practice summarised the position as follows:

In the absence of ... settled EPO practice or case-law, and bearing in mind the views of the Court of Appeal expressed in both *Symbian* and *Aerotel*, the Office is not bound to follow the EPO practice. In any event, the Office remains bound by the precedents set by the UK Courts. Consequently, the assessment of whether an invention is no more than a computer program as such is set out in the Court of Appeal’s decisions in *Aerotel* and *Symbian*.⁵⁴²

⁵³⁹ G3/08 *Programs for computers* (Opinion of 12 May 2010) at [10.13].

⁵⁴⁰ G3/08 *Programs for computers* (Opinion of 12 May 2010) at [7.2]

⁵⁴¹ Discussed above at 138.

⁵⁴² United Kingdom Patent Office, “1. Patentable Inventions” *Manual of Patent Practice – Patents Act 1977*, updated 1 July 2010 <<http://www.ipo.gov.uk/practice-sec-001.pdf>> (13 July 2010) at [1.29.7].

The United States

In re Comiskey

In *Comiskey*,⁵⁴³ the Federal Circuit dealt with claims to a “method and system for mandatory arbitration resolution regarding legal documents, such as wills or contracts”.⁵⁴⁴ The Court held that claims which sought to patent “the use of human intelligence in and of itself”⁵⁴⁵ were non-patentable.

In rejecting the claims, the court drew attention to “machine or transformation” test sourced in *Diehr*. But considered that this approach was consistent with *State Street*.

It was further suggested in relation to the remaining claims that the “routine addition of modern electronics to an otherwise unpatentable invention typically creates a *prima facie* case of obviousness”.⁵⁴⁶ These claims were remanded to the USPTO for further consideration.

In re Bilski

In *Bilski*,⁵⁴⁷ an *en banc* panel of the Court of Appeals for the Federal Circuit marked a sharp change in direction, affirming the rejection of a claim for “a method of hedging risk in the field of commodities trading”⁵⁴⁸ as not directed to patentable subject matter. In doing so, the Court also held that the *State Street* “useful, concrete or tangible result” test was “insufficient to determine whether a claim is patent-eligible under §101.”⁵⁴⁹ The Court was concerned to ensure that “a process claim is tailored narrowly enough to encompass only a particular application of a fundamental principle rather than to pre-empt the principle itself.”⁵⁵⁰ To do so, the Court noted that a claim will only be patent-eligible where:

- (1) it is tied to a particular machine or apparatus; or
- (2) it transforms a particular article into a different state or thing.⁵⁵¹

The Court further explained that this

machine-or-transformation test is a two-branched inquiry; an applicant may show that a process claim satisfies § 101 either by showing that his claim is tied to a particular machine, or by showing that his claim transforms an article. Certain considerations are applicable to analysis under either branch First, . . . the use of a specific machine or transformation of an article must impose meaningful limits on the claim’s scope to impart patent-eligibility. Second, the involvement of the

⁵⁴³ *In re Comiskey* 499 F.3d 1365 (2007).

⁵⁴⁴ *In re Comiskey* 499 F.3d 1365 (2007) at 1368.

⁵⁴⁵ *In re Comiskey* 499 F.3d 1365 (2007) at 1379.

⁵⁴⁶ *In re Comiskey* 499 F.3d 1365 (2007) at 1380.

⁵⁴⁷ *In re Bernard L. Bilski and Rand A. Warsaw* (2008) 545 F.3d 943.

⁵⁴⁸ *In re Bilski* 545 F.3d 943 (2008) at 949.

⁵⁴⁹ *In re Bilski* 545 F.3d 943 (2008) at 959.

⁵⁵⁰ *In re Bilski* 545 F.3d 943 (2008) at 954.

⁵⁵¹ *In re Bilski* 545 F.3d 943 (2008) at 954.

Chapter 3: How Software Became Patentable

machine or transformation in the claimed process must not merely be insignificant extra-solution activity.⁵⁵²

The approach of the majority also suggests that the doctrine of mental steps may be having a revival, since “a claimed process wherein all of the process steps may be performed entirely in the human mind is obviously not tied to any machine and does not transform any article into a different state or thing. As a result, it would not be patent-eligible under § 101.”⁵⁵³

The impact of such a test on “pure” business methods is clear:

Purported transformations or manipulations simply of public or private legal obligations or relationships, business risks, or other such abstractions cannot meet the test because they are not physical objects or substances, and they are not representative of physical objects or substances.⁵⁵⁴

At the same time, the court left open the issue of whether claims tied to a computer were sufficient, preferring instead to leave “to future cases the elaboration of the precise contours of machine implementation,”⁵⁵⁵ including “whether or when recitation of a computer suffices to tie a process claim to a particular machine.”⁵⁵⁶

Bilski v Kappos

The Federal Court’s *Bilski* decision was appealed to the Supreme Court, and oral arguments were heard on 9 November 2009. On 28 June 2010, the US Supreme Court handed down its long anticipated decision in *Bilski v Kappos*.⁵⁵⁷ It was expected before the decision was released that Stevens J, who wrote the opinion of the Court in *Flook* and strongly dissented in *Diehr* would write the majority opinion. However, the opinion of the Court was delivered by Kennedy J, with a split along conservative-liberal lines being evident in the 5-4 majority.

Agreement As the existence of three concurring judgments suggests, there was a great deal of agreement between the members of the Court. The points of agreement were helpfully set out in summary in the judgment of Breyer J:

1. That “the claims at issue here are unpatentable abstract ideas”.⁵⁵⁸
2. That “although the text of §101 is broad, it is not without limit”.⁵⁵⁹ To this end it was pointed out that “[p]henomena of nature, though just discovered, mental processes, and

⁵⁵² *In re Bilski* 545 F.3d 943 (2008) at 961-962.

⁵⁵³ *In re Bilski* 545 F.3d 943 (2008) at 961, footnote 26. This reasoning appears to have been recently endorsed in *CyberSource Corporation v Retail Decisions Inc* Appeal No 2009-1358 (Fed Cir, 2011).

⁵⁵⁴ *In re Bilski* 545 F.3d 943 (2008) at 963.

⁵⁵⁵ *In re Bilski* 545 F.3d 943 (2008) at 962.

⁵⁵⁶ *In re Bilski* 545 F.3d 943 (2008) at 962.

⁵⁵⁷ *Bilski v Kappos* 130 S. Ct. 3218 (2010).

⁵⁵⁸ *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3258 per Breyer J.

⁵⁵⁹ *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3258 per Breyer J.

abstract intellectual concepts are not patentable” under §101, since allowing individuals to patent these fundamental principles would ‘wholly pre-empt’ the public’s access to the ‘basic tools of scientific and technological work.’”⁵⁶⁰

3. That although “the so-called ‘machine-or-transformation test’ has ... repeatedly helped the Court to determine what is ‘a patentable ‘process’”,⁵⁶¹ “it has never been the ‘sole test’ for determining patentability”.⁵⁶² It remains an “important example of how a court can determine patentability under §101, but the Federal Circuit erred in this case by treating it as the *exclusive test*”.⁵⁶³ The test was also described by Breyer J as “a useful and important clue”⁵⁶⁴ to patentability, that “the Court intends neither to de-emphasize the test’s usefulness nor to suggest that many patentable processes lie beyond its reach.”⁵⁶⁵
4. The rejection of the machine-or-transformation test as the sole test, “by no means indicates that anything which produces a ‘useful, concrete and tangible result’ ... is patentable.”⁵⁶⁶ Breyer J cited Stevens J in support⁵⁶⁷ where his Honour had said that it would be a “grave mistake” to assume the continued applicability of the *State Street* “useful, concrete and tangible result” test. Thus although the Kennedy J’s opinion was neutral on the *State Street* test, it can safely be considered that a rejection of the *State Street* test represents a majority view.

Disagreement

Kennedy J opinion The structure of the judgment itself shows a division on the issue between the members of the Court. There was a 5-4 division along conservative/liberal lines, with Scalia J having a foot in both camps, in that his Honour did not join the majority decision *in toto*, and instead joined Breyer J (who agreed with Stevens J) in respect of part of the judgment.

The main source of disagreement was as to the existence of a business method exception, with the majority holding that the “ordinary, contemporary, common meaning”⁵⁶⁸ of the word “process” in the patent statute necessarily encompassed some business methods.⁵⁶⁹ This position was further fortified, it was said, by the provision in §273 of a “prior use” defence to infringement for business method claims.

The majority held that instead of relying on a new test for patentability, the preferable approach was to look to the “three specific exceptions to §101’s broad principles: laws of nature, physical

⁵⁶⁰ *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3258 per Breyer J, citing in support *Gottschalk v Benson* 409 US 63, 67, 72 (1972); *Diamond v Diehr* 450 US 175, 185 (1981) and *Diamond v Chakrabarty* 447 US 303, 309 (1980).

⁵⁶¹ *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3258 per Breyer J, citing in support *Parker v Flook* (ref)

⁵⁶² *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3259 per Breyer J.

⁵⁶³ *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3259 per Breyer J (original emphasis).

⁵⁶⁴ *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3258 per Breyer J.

⁵⁶⁵ *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3259 per Breyer J.

⁵⁶⁶ *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3259 per Breyer J. See also *ibid*: “To the extent that the Federal Circuit’s decision in this case rejected that approach, nothing in today’s decision should be taken as disapproving of that determination.”

⁵⁶⁷ *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3259 per Breyer J, citing Stevens J at 3232, footnote 1.

⁵⁶⁸ *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3221 per Kennedy J.

⁵⁶⁹ This same textual approach was the basis for rejecting the machine-or-transformation test as the sole test.

Chapter 3: How Software Became Patentable

phenomena and abstract ideas⁵⁷⁰ as developed in *Benson*, *Flook* and *Diehr*, which established a sufficient basis upon which to hold Bilski's invention was an unpatentable abstract idea. As such it was not necessary to go any further in defining what amounts to a patentable process. It is on this basis that one commentator declared the majority opinion to be "business as usual".⁵⁷¹

The plurality part of Kennedy J's judgment expressed concern for the application of historical tests to emerging technologies, on the basis that "times change"⁵⁷² and "new technologies may call for new enquiries"⁵⁷³ and seems to have suggested that inventions of this Information Age are presumably patent-eligible: "the machine-or-transformation test would create uncertainty as to the patentability of software, advanced diagnostic medicine techniques, and inventions based on linear programming, data compression, and the manipulation of digital signals."⁵⁷⁴

Stevens J concurrence Stevens J was critical of textual approach of the majority, noting that "if this portion of the Court's opinion were taken literally, the results would be absurd: Anything that constitutes a series of steps would be patentable so long as it is novel, nonobvious, and described with specificity."⁵⁷⁵ His Honour then noted that the rejection of the business method exception and machine-or-transformation tests as "atextual" could not be reconciled with the majority's finding that such a test was "useful and important"⁵⁷⁶ nor "when the Court excludes processes that tend to pre-empt commonly used ideas"⁵⁷⁷ as it did in this case. Further, a purely literal reading of §101 would preclude the Court's holding that claims to laws of nature, natural phenomena and abstract ideas "do not count as processes ... even if they can be colloquially described as such."⁵⁷⁸ As such his Honour noted that although a textual approach is "a fine approach to statutory interpretation in general, it is a deeply flawed approach to a statute that relies on complex terms of art developed against a particular historical background."⁵⁷⁹

Stevens J was also critical of the lack of "a satisfying account of what constitutes an unpatentable abstract idea. ... The Court essentially asserts its conclusion that petitioners' application claims an abstract idea. This mode of analysis (or lack thereof) may have led to the correct outcome in this case, but it also means that the Court's musings on this issue stand for very little."⁵⁸⁰

Instead, his Honour preferred to "restore patent law to its constitutional and historical moorings".⁵⁸¹ As such his Honour traced the development of US patent law, back to the English law upon which it was based. In a detailed history of patentable subject matter,⁵⁸² his Honour set

⁵⁷⁰ *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3221 per Kennedy J, quoting *Diamond v Chakrabarty* 447 US 303 at 309.

⁵⁷¹ Dennis Crouch, "Bilski v Kappos" *Patently-O*, 29 June 2010 <<http://www.patentlyo.com/patent/2010/06/bilski-v-kappos-business-methods-out-software-still-patentable.html>> (30 June 2010).

⁵⁷² *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3227 per Kennedy J.

⁵⁷³ *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3228 per Kennedy J.

⁵⁷⁴ *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3227 per Kennedy J.

⁵⁷⁵ *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3235 per Stevens J.

⁵⁷⁶ *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3235 per Stevens J.

⁵⁷⁷ *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3235 per Stevens J.

⁵⁷⁸ *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3238-3239 per Stevens J (footnote omitted).

⁵⁷⁹ *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3238 per Stevens J.

⁵⁸⁰ *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3236 per Stevens J.

⁵⁸¹ *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3232 per Stevens J.

⁵⁸² *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3239-3250 per Stevens J.

out how “for centuries, it was considered well established that a series of steps for conducting business was not, in itself, patentable”,⁵⁸³ and that the enactment of the 1952 represented a legislative codification of this understanding.

As to the existence §273 defence, Stevens J pointed out that this 1999 amendment formed “a hazardous basis for inferring the intent of”⁵⁸⁴ the 1952 Congress in passing §101. If anything, the §273 was a Congressional response to concerns about the impact of such a development on the business community which if anything suggested a lack of Congressional support for the patenting of business methods, reflecting instead “surprise and perhaps even dismay that business methods might be patented.”⁵⁸⁵ As such, §273 was “a red herring; we should be focusing our attention on §101 itself.”⁵⁸⁶

In summary his Honour noted that

[t]he scope of patentable subject matter ... is broad. But it is not endless. In the absence of any clear guidance from Congress, we have only limited textual, historical, and functional clues on which to rely. Those clues all point towards the same conclusion: that the petitioners’ claim is not a “process” within the meaning of §101 because methods of doing business are not, in themselves, covered by the statute. In my view acknowledging as much would be a far more sensible and restrained way to resolve this case. Accordingly, while I concur in the judgment, I strongly disagree with the Court’s disposition of this case.⁵⁸⁷

If nothing else, the complexity of, and division within, the *Bilski v Kappos* judgment highlights the difficult nature of the patentable subject matter inquiry. Although many had hoped that *Bilski* would be dispositive of the issues, it is clear from the judgment that further work will need to be done. This seems to be the preferred solution, with the Supreme Court inviting the Federal Circuit’s “development of other limiting criteria that further the purposes of the Patent Act and are not inconsistent with its text.”⁵⁸⁸

Impact on US law Whilst the ramifications of the case are still being worked out, an early indication as to the likely development of law is provided by the interim guidelines developed by the US Patent and Trademark Office.⁵⁸⁹ The Guidelines repeat the position that the machine-or-transformation test, although not the sole test remains a “useful and important clue”,⁵⁹⁰ and further note that “[t]o date, no court, presented with a subject matter eligibility issue, has ever ruled that a method claim that lacked a machine or a transformation was patent-eligible.”⁵⁹¹

⁵⁸³ *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3232 per Stevens J.

⁵⁸⁴ *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3250 per Stevens J.

⁵⁸⁵ *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3251 per Stevens J.

⁵⁸⁶ *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3252 per Stevens J.

⁵⁸⁷ *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3257 per Stevens J.

⁵⁸⁸ *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3231 per Kennedy J.

⁵⁸⁹ United States Patent and Trademark Office, “Interim Guidelines for Determining Subject Matter Eligibility Claims in View of *Bilski v Kappos*” (2010) 75(143) *Federal Register* 43,922 (“USPTO Interim Guidelines”)

⁵⁹⁰ United States Patent and Trademark Office, “Interim Guidelines for Determining Subject Matter Eligibility Claims in View of *Bilski v Kappos*” (2010) 75(143) *Federal Register* 43,922 at 43,924.

⁵⁹¹ United States Patent and Trademark Office, “Interim Guidelines for Determining Subject Matter Eligibility Claims in View of *Bilski v Kappos*” (2010) 75(143) *Federal Register* 43,922 at 43,924.

Chapter 3: How Software Became Patentable

The Guidelines also expand on the approach to be taken to examination where the abstract idea exception may apply, namely “where a claimed ‘method’ [does] not sufficiently recite a physical instantiation”.⁵⁹² As such, the Guidelines identify a number of “factors that are to be considered when evaluating patent-eligibility of method claims.”⁵⁹³ Those factors are:

- (a) Whether the method involves or is executed by a particular machine or apparatus. If so, the claims are less likely to be drawn to an abstract idea; if not, they are more likely to be so drawn. . . .
- (b) Whether performance of the claimed method results in or otherwise involves a transformation of a particular article. If such a transformation exists, the claims are less likely to be drawn to an abstract idea; if not, they are more likely to be so drawn. . . .
- (c) Whether performance of the claimed method involves an application of a law of nature, even in the absence of a particular machine, apparatus, or transformation. If such an application exists, the claims are less likely to be drawn to an abstract idea; if not, they are more likely to be so drawn. . . .
- (d) Whether a general concept (which could also be recognized in such terms as a principle, theory, plan or scheme) is involved in executing the steps of the method. The presence of such a general concept can be a clue that the claim is drawn to an abstract idea. . . .⁵⁹⁴

There have been no major upheavals in the law since the Guidelines were published, although there have been some developments in accordance with those Guidelines. In particular, it is noted that the decision in *CyberSource Corporation v Retail Decisions Inc*⁵⁹⁵ held that claims to software on a carrier (*Beauregard* claims) fall afoul of the machine-or-transformation “clue” and are therefore not properly addressed to patentable subject matter. Also, the decision in *Association for Molecular Pathology v USPTO*,⁵⁹⁶ demonstrates the application of the post-*Bilski* approach by the Federal Circuit to biotechnology subject matter, holding that claims to isolated gene sequences are patentable, but diagnostic methods that only compare or analyze sequences are not.

Impact on Australian law The *Bilski* decision puts Australian patent law on the horns of a familiar dilemma. Having tracked US case law, and established a reliance on a modified version of the *State Street* test,⁵⁹⁷ we are now out on our own. A textual interpretation of the scope of patentable subject matter by reference to the ordinary meaning of the word “process”

⁵⁹² United States Patent and Trademark Office, “Interim Guidelines for Determining Subject Matter Eligibility Claims in View of *Bilski v Kappos*” (2010) 75(143) *Federal Register* 43,922 at 43,924.

⁵⁹³ United States Patent and Trademark Office, “Interim Guidelines for Determining Subject Matter Eligibility Claims in View of *Bilski v Kappos*” (2010) 75(143) *Federal Register* 43,922 at 43,924.

⁵⁹⁴ United States Patent and Trademark Office, “Interim Guidelines for Determining Subject Matter Eligibility Claims in View of *Bilski v Kappos*” (2010) 75(143) *Federal Register* 43,922 at 43,925 to 43,926.

⁵⁹⁵ Appeal No 2009-1358 (Fed Cir, 2011).

⁵⁹⁶ Appeal No 2010-1406 (Fed Cir, 2011).

⁵⁹⁷ *Grant v Commissioner of Patents* (2006) 154 FCR 62 at 70 notes the absence of an “artificial state of affairs, in the sense of a concrete, tangible, physical, or observable effect concrete, tangible, physical, or observable effect”, which is derived from US case law, at 68.

is of no value in the Australian context since the Australian act is not phrased in such terms.⁵⁹⁸ In any event, the legislative reliance on the judicial development of the concept of manner of manufacture since the *Statute of Monopolies* means that our understanding of patentable subject matter is firmly entrenched in history.

Thus the judgment which best suits the Australian approach is in fact the Stevens J concurrence which sources the patentable subject matter inquiry in its history. In particular, his Honour's discussion of the distinction between useful and fine arts as understood at the time of the passage of the original act represents a useful review of principles which are of direct relevance to the Australian approach in light of the express reliance on the distinction in *NRDC*.

Overview of recent developments

Overall then, the recent developments across the three jurisdictions considered in this chapter evidence the ongoing struggle which the patentable subject matter inquiry represents. Neither the UK nor US positions evidence any form of settled approach to determining the scope of inherent patentability any time soon.

However, putting aside the jurisprudence of the EPO, which seems to be on a different path, and the UK position which has changed little since *Aerotel*, it is submitted that the US and Australian positions share a common thread, namely that the pendulum seems to be swinging away from broad notions of patentability. Whilst the case law of previous eras celebrated the growth of the scope of patentable subject matter, the recent decisions highlight that "whilst [it] is broad, it is not without limit".⁵⁹⁹ How to define those limits is the source of disagreement.

5 Conclusion

This chapter has charted the journey of software towards its initial classification as non-patentable subject matter, up to the current approach to patentability in Australia, the United Kingdom and the United States. Although the road to the patentability of software has been understandable, the problems which the permissive approach to characterisation of claims in this context creates are far from resolved.

Broadly, the development of the case law in each of the three jurisdictions illustrates two propositions. Firstly, that software and software-related inventions sit right at the borderline of patentable subject matter. Secondly, that defining that borderline is a difficult issue indeed. It is also submitted that the case law demonstrates the difficulty of building a solid rationale on legal decisions which are tightly intertwined with the facts on which they are based. It is perhaps as a result of this factual dependence that the cyclical nature of under- and over-protection comes into being.

⁵⁹⁸ Note however that *Grant* consistently held that the claimed invention amounted to an "abstract, intangible situation": at 70, and was therefore non-patentable. This is consistent with the unanimous holding of the *Bilski* judges that abstract ideas are non-patentable. See below.

⁵⁹⁹ *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3258 per Breyer J.

Chapter 3: How Software Became Patentable

In Australia these difficulties are apparent in the *Grant* decision where the court sought awkwardly to bring together prior case law into a workable reinterpretation of *NRDC*. The fact remains that the threshold of patentable subject matter is necessarily difficult, and will never be as simple as coming up with a new and revised formula. It seems like there has been a period of over-protection in which patent law has strayed too far into the realm of abstractions.

The next chapter will look at the reasons for these continued practical difficulties, linking them back to the nature of software as first discussed in Chapter 1. This exploration will begin the larger exploration of how to properly structure the patentability inquiry so that it can properly handle the patentability of software and software-dependent inventions in a rational and consistent fashion

Why Software Patents are a Problem

“The chance that a law will achieve its intended purpose improves when it is grounded in an accurate understanding of the phenomena it will regulate.”¹

This chapter will explain the software patent problem, by defining what a software patent is, and why it is that software patents are such a difficult fit within the patent regime, tracing these difficulties to the key considerations identified in Chapter 1.² First it was noted that software has an abstract, intangible nature, as opposed to the hardware on which it runs. As such it is important to consider context, namely, the type of software component being developed and how many layers of abstraction sit between it and the hardware. Is it a part of the operating system, and thus closely linked to hardware, or is it a high-level platform-independent application component? Secondly it was noted that reuse is critical in software development, since new software innovations almost inevitably build on what has come before. Finally, software products are highly complex, involving millions of discrete components. The way in which these factors combine to produce the software patent problem is set out below.

The chapter will also document the way in which the courts of the US, the EU and Australia have attempted to deal with these problems, and describe why it is that the software patent issue should be dealt with as a patentable subject matter issue, rather than as an issue of novelty or non-obviousness.

It will be shown why the current approaches are insufficient, and to introduce the notion that a fresh approach to the software patent problem is needed.

1 What is a software patent?

One difficulty which must be overcome before the software patent problem can be usefully discussed is that there is no agreed definition of what a software patent is. A broad definition of software might include anything that is not hardware, hardware being the “physical

¹ Pamela Samuelson, “Foreword: The Digital Content Symposium” (1997) 12(1) *Berkeley Technology Law Journal*. <<http://www.law.berkeley.edu/journals/btlj/articles/vol12/Samuelson/html/reader.html>> (25 February 2008).

² See Chapter 1, at 22.

Chapter 4: Why Software Patents are a Problem

component of a computer system”,³ or the “part[s] of a computer that is fixed and cannot be altered without replacement or physical modification”.⁴ Following on from this definition then, a software patent would be a patent for a computer-based invention which can be altered without replacement or physical modification. An alternative definition is that offered by Allison and Lemley, who define a software patent as “[a]n invention that is completely embodied in software, even if the claims of the patent refer to a system or article of manufacture.”⁵ Such a definition would exclude inventions which include software as a component of a physical device. The difficulty with such a definition is twofold. Firstly, it underestimates the extent to which the physical elements of a computer-based innovation can directly limit related software components. A good example is firmware, discussed in Chapter 1, which is embedded in a hardware device, and is generally limited in use to interaction with the particular physical components of a device (so largely tied to it), but can be modified via a “flash upgrade” and thus has the modifiability of software. Secondly, by focusing on ‘pure’ software patents such as this, one runs the risk of failing to consider patents which have been drafted to appear as physical devices in which software is merely a component.⁶

Thus a proper discussion of software patents should consider as wide a range of software and software-related innovations as possible. It is preferable to take a broader view of what amounts to a software patent, namely that put forward by Bessen and Hunt:

Our concept of software patent involves a logic algorithm for processing data that is implemented via stored instructions; that is, the logic is not “hard-wired.” These instructions could reside on a disk or other storage medium or they could be stored in “firmware,” that is, a read-only memory, as is typical of embedded software.⁷

2 Why is the patenting of software problematic?

The different historical origins of the modern patent regimes in Australia, the US and UK have resulted in very different approaches to the issue of patentable subject matter. Although Australia and the UK have shared history over a period of hundreds of years, the adoption of the European Patent Convention (EPC) by the UK in 1977 forever broke that link. Whereas the Australian position continues to define patentable subject matter by reference to the definition in the *Statute of Monopolies 1623*, and the notion therein of a manner of manufacture, the UK position is now defined by the express exclusions in Article 52 of the EPC, and their interpretation by the European Patent Office (EPO) as resolving to a single requirement of technical character. The US approach, although likely influenced by the *Statute of Monopolies*, derives

³ “Hardware” in M J Clugston et al (eds), *The New Penguin Dictionary of Science* (Penguin, 2004).

⁴ “Hardware” Wiktionary <en.wiktionary.org/wiki/hardware> (6 November 2008).

⁵ John R Allison and Mark A Lemley, “Who’s Patenting What? An Empirical Exploration of Patent Prosecution” (2000) 58 *Vanderbilt Law Review* 2099 at 2110.

⁶ See the discussion below in Section 2.2, where claims have been directed to apparatus executing the software, or to a physical carrier containing the executable software. See also James E Bessen and Robert M Hunt, “An Empirical Look at Software Patents” (2007) 16(1) *Journal of Economics & Management Strategy* 157 at 163.

⁷ James E Bessen and Robert M Hunt, “An Empirical Look at Software Patents” (2007) 16(1) *Journal of Economics & Management Strategy* 157 at 163.

Why is the patenting of software problematic?

uniquely from the power afforded to Congress in the US Constitution to pass laws to “promote the Progress of Science and useful Arts”⁸, and is determined by the four statutory classes of patentable subject matter defined in 35 USC §101 namely, processes, machines, manufactures and compositions of matter.

Despite significant differences in the approach, there a number of similarities. Put simply, software has been found to be patentable subject matter in all three jurisdictions, even despite the express exclusion of computer programs under the EPC. This end result derives from the approach of courts and tribunals to the characterisation of software patent claims, and assumptions therein about the nature of software. It will now be shown how these approaches have contributed to an essentially identical software patent problem, which manifests itself both in theoretical problems for patent law and practical problems for the software industry. These problems will be considered by reference to the four key elements of the nature of software noted above, namely abstraction, context, complexity and reuse.

Abstraction

As was discussed in Chapter 1, the history of software has been one of increasing layers of abstraction between software and the hardware which runs it. All but firmware and the lowest levels of the operating system are implemented largely, or entirely independently, of the physical machine on which they are run. Some software components, towards the top of the current stack of abstractions are very far removed from their physical effect indeed.

Thus there is a tension with one of the classic doctrines of patent law noted in Chapter 2, namely that patents are supposed to protect the execution or application of an idea, in the form of an invention, not the idea itself.⁹ When the invention has a physical manifestation, such as a new machine, it is easy to distinguish the invention from the idea behind it. However, patent law long outgrew such simple inventions, with the move to ever more abstract subject matter beginning with patents for improvements to machines and devices, through methods of effecting such improvements, and new uses of existing machines and substances,¹⁰ to the “super-abstract” information-based innovations of the present age.

The distinction put forward in *Hickton’s* case,¹¹ which only requires that “a way of carrying [the idea] out”¹² before an abstract idea becomes a patentable invention, is particularly problematic. It will be recalled from Chapter 1 that the software development process generally involves a gradual move from the idea behind a program towards the more specific code actually implementing it. Knowing at which point the non-patentable idea has been expressed in sufficiently

⁸ United States Constitution, Article One, Section 8, Clause 8.

⁹ A non-patentable or “abstract” idea (see *O’Reilly v Morse* 56 US 62 (1852), Chapter 2 at 48) has also been described as a principle, (*Boulton v Bull* (1795) 126 ER 651, Chapter 2 at 42; *Le Roy v Tatham* 55 US 156 (1852), Chapter 2 at 48); and an “abstract notion” (*Boulton v Bull* (1795) 126 ER 651).

¹⁰ See the discussion of the non-patentability of analogous uses in Chapter 2 at 47 and following; see also the discussion of analogous uses in *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 (Chapter 2 at 61).

¹¹ *Hickton’s Patent Syndicate v Patents and Machine Improvements Co. Ltd* (1909) 26 RPC 339 (Chapter 2 at 52).

¹² *Hickton’s Patent Syndicate v Patents and Machine Improvements Co Ltd* (1909) 26 RPC 339 at 348 per Fletcher-Moulton LJ.

Chapter 4: Why Software Patents are a Problem

concrete terms so as to amount to an invention is a difficult task. What was already “a question of degree”¹³ may seem to be one of splitting hairs, encouraging the abandonment of the distinction altogether.¹⁴

The same problem is faced when the distinction sought to be made is between a discovery (or idea) and its practical application,¹⁵ where the suggested application is itself abstract in nature.¹⁶ Such patents, due to their breadth, are prone to giving the patentee control over a wide range of independently developed technologies – a windfall for the patent holder at the expense of other innovators. As a result, the general public suffer from the slowing of “the onward march of science”.¹⁷

If the history of the computer industry set out in Chapter 1 continues to develop in the same way,¹⁸ these problems will only be exacerbated. Further abstractions will continue to be layered on top of each other, bringing programming languages ever closer to natural language, and making it harder to discern the difference between programming and user interaction.¹⁹

This is not to say that courts are unaware of the dangers of over-protection. A range of responses to prevent the award of patents “so abstract and sweeping as to cover both known and unknown uses”²⁰ have been developed. However, it will now be shown how the approaches adopted have failed to provide an effective safeguard.

¹³ *Harwood v Great Northern Railway* (1865) 11 ER 1488 at 1499 (Chapter 2 at 47).

¹⁴ This is arguably what happened in recent European Patent Office jurisprudence in which the computer program exclusion has been reduced to a mere form provision. An overview of this “any hardware” approach is set out in the discussion of the case *Aerotel Ltd v Telco Holdings Ltd and in the matter of Macrossan’s Application* [2006] EWCA Civ 1371; [2007] RPC 7 (Chapter 3 at 131). A more detailed discussion of these developments is set out in Chapter 3 in Section 3 on page 115. See also section 4 below.

¹⁵ See for example *Gottschalk v Benson* 409 US 63 (1973) (Chapter 3 at 86) where it was held that allowing the claim in the absence of a specific practical application would amount to a patent on an idea; see also *Parker v Flook* 437 U.S. 584 (1978) (Chapter 3 at 88) where it was held that the required application must be more than insignificant post-solution activity; *Bilski v Kappos* 130 S. Ct. 3218 (2010) (Chapter 3 at 142) applied these approaches in holding the subject claims to be a non-patentable abstract idea. Cf *Burroughs Corporation’s Application* [1973] FSR 439 (Chapter 3 at 101) at 449 where a “naked conception” or “bare method or idea” was contrasted with “practical embodiment” of software “enabling that method to be realised in practice”. Similarly in *Genentech Inc’s Application (Human Growth Hormone)* [1989] RPC 147 (Chapter 3 at 107) at 240 it was held that “the practical application of a discovery” was patentable subject matter since such a claim does not relate to the discovery “as such”. In the Australian context, see *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 (Chapter 2 at 61) at 264, where the court characterised a non-patentable discovery as “abstract information without any suggestion of a practical application”.

¹⁶ That this context is important consideration is further developed below.

¹⁷ *Gottschalk v Benson* (1973) 409 US 63 at 68 (Chapter 3 at 86).

¹⁸ This seems probable, at least in the short to medium term, absent some disruptive new methodology superseding the existing approach.

¹⁹ Some significant headway has already been made in such areas as speech recognition and in building systems to parse written language. But as anyone who has used interactive voice recognition menus from their telephone knows, the current state of the art is far from perfect. For an interesting opinion on the current (allegedly fatal) problems facing speech recognition technology, see Robert Fortner, “Rest in Peas: The Unrecognized Death of Speech Recognition” <<http://robertfortner.posterous.com/the-unrecognized-death-of-speech-recognition>> (2 August 2010). For a practical introduction to the difficulties in teaching computers to understand natural language see Peter Bradley, “Turing Machines and Natural Language” *Consortium on Cognitive Science Instruction*, 2002 <http://www.mind.ilstu.edu/curriculum/turing_machines/turing_machines_and_language.php> (2 August 2010).

²⁰ *Gottschalk v Benson et al* (1973) 409 U.S. 63 at 68.

Pre-emption/breadth

In the UK, the House of Lords in *British United*²¹ noted that a claim might, because of its breadth be “not proper subject matter”,²² since a claim ought to “inform others of the limits of such protection”.²³ A similar view was expressed in *RCA Photophone*²⁴ where this requirement was held to be the “consideration” for the monopoly.²⁵ It was also noted that an overbroad claim also creates “a public nuisance”.²⁶ A similar limitation arises from the judgment of Parker J in *British United*, where his Honour asserted that no authority existed for the proposition that the discoverer of a new principle “could have protected himself against all means of solving [a] problem”.²⁷ This line of reasoning also appears to inform the reasoning in other subsequent case law,²⁸ although was subjected to a complicating and largely unhelpful caveat in *David Kahn*²⁹ that such subject matter might not in fact be excluded where “invention lies in identification of the problem”.³⁰

Along similar lines in the US, the Supreme Court in *Gottschalk v Benson* limited claims involving mathematical algorithms where such claims, due to the absence of a practical application would “wholly pre-empt the mathematical formula and in practical effect would be a patent on the algorithm itself”.³¹ Although subsequent interpretation of this requirement lead to an unsatisfactorily narrow requirement that such claims first be directed to a mathematical formula or its equivalent,³² the US Supreme Court in *Bilski v Kappos* has brought this pre-emption issue back to the fore.³³

The main problem with this approach is that by only looking to the practical effect of a patent to determine whether a patent should be available in a particular case, no overarching principles can be set out. This overlooks the potential usefulness of considering exclusions at the categorical level, in that the nature of certain technologies gives rise to policy considerations which are likely to affect. It also fails to acknowledge that categorical exclusions have been part of patent law since at least the passage of the *Statute of Monopolies*.³⁴ Further, the starting point in this

²¹ *British United Shoe Machinery Co Ltd v Simon Collier Ltd* (1909) 26 RPC 21 (Chapter 2 at 53).

²² *British United Shoe Machinery Co Ltd v Simon Collier Ltd* (1909) 26 RPC 21 at 50.

²³ *British United Shoe Machinery Co Ltd v Simon Collier Ltd* (1909) 26 RPC 21 at 50.

²⁴ *RCA Photophone Ltd v Gaumont-British Picture Corporation* (1936) 53 RPC 167 at 186-187. The case was discussed in Chapter 2 at 54.

²⁵ This is of course informed by the view of patent law as a social contract. See the discussion of the social contract in the context of *RCA Photophone Ltd v Gaumont-British Picture Corporation* (1936) 53 RPC 167 (Chapter 2 at 54); and the discussion of *Liardet v Johnson* (1778) 1 Carp Pat Cas 35 (NP) (Chapter 2 at 45).

²⁶ *RCA Photophone Ltd v Gaumont-British Picture Corporation* (1936) 53 RPC 167 at 194.

²⁷ *British United Shoe Machinery Co Ltd v Simon Collier Ltd* (1909) 26 RPC 21 at 50.

²⁸ See for example *Gale's Patent Application* [1991] RPC 305 at 327-28 where it was held that the claimed invention was not patentable, *inter alia*, because it did not “solve a ‘technical’ problem lying within the computer”. The full quote is extracted in Chapter 3 at 110.

²⁹ *David Kahn Inc v Conway Stewart* [1974] RPC 279, discussed in Chapter 2 at 53, footnote 117.

³⁰ *David Kahn Inc v Conway Stewart & Co Ltd* (No 2) [1974] RPC 279 at 319-320.

³¹ *Gottschalk v Benson* 409 US 63 (1973) at 72. This concern also informs the rejection of insignificant post-solution activity in *Parker v Flook* 437 US 584 (1978). See also *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3230 per Kennedy J.

³² See *In re Freeman* 573 F.2d 1243 (1978) at 1246 (Chapter 3 at 3).

³³ See *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3239 per Kennedy J: “[T]he Court resolves this case narrowly on the basis of this Court’s decisions in *Benson*, *Flook* and *Diehr*, which show that petitioners’ claims are not patentable...”.

³⁴ See for example Coke’s explanation of the phrase “contrary to law” in his *Institutes of Law*, discussed in Chapter 2 at 2, as excluding improvements to existing manufactures, an exclusionary ground which Coke traces back to *Bircot’s case*, decided in 1572.

Chapter 4: Why Software Patents are a Problem

analysis is a general presumption that patents should be available, with exceptions for particular instances. The policy choices which inform such a presumption are never aired. Much the same criticism can be made of the mere identification of breadth and pre-emption as the source of the problem. The exact harm which these doctrines seek to overcome is never specified. Why is it for example that a claim to an algorithm *should* be considered non-patentable? Put simply, the case law just discussed “never provides a satisfying account of what constitutes [an overbroad claim]”³⁵ and “essentially asserts [a] conclusion”.³⁶

Mental steps doctrine

Another approach adopted in the US and detailed in Chapter 3, was to disallow patentability under the “mental steps doctrine”.³⁷ This doctrine recognised that claims directed to highly abstract subject matter, as “intangible, illusory and non-material things”³⁸ run an appreciable risk of infringement by human operators carrying out the claimed process. Whilst it was generally agreed that patents should not be awarded for mental steps, the application of the doctrine becomes more complicated where the steps involve a combination of “positive and physical steps as well as so-called mental steps”³⁹ since the involvement of human operators had previously been considered unproblematic. Cases such as *Abrams*⁴⁰ held that the reasons for the exception were “self-evident”⁴¹ although the lack of a solid theoretical underpinning was to prove its downfall.⁴²

The doctrine is complicated in the software context due to the intuitive problem with holding that claims directed to a machine could be reasonably interpreted as reading upon human mental activity. The notion that a human mind operated in the same way as a computer was explicitly rejected in *Bernhart*,⁴³ and held to be beyond a “reasonable interpretation of the claims” in the decision of the Court of Customs and Patent Appeals (CCPA) in *Benson*.⁴⁴ In that case the fact that a computer could carry out the process “without human intervention”⁴⁵ was sufficient

³⁵ *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3236 per Stevens J.

³⁶ *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3236 per Stevens J.

³⁷ This is not to say that its relevance is limited to that jurisdiction. As noted in Chapter 3, the EPC excludes the patentability of “schemes, rules and methods for performing mental acts ... and programs for computers” which might suggest a similar relationship between the two. Certainly the examiner in *Fujitsu* [1996] RPC 511 (Chapter 3 at 111) rejected the claimed invention as falling within one or both exclusions. Laddie J held that the computer program exclusion had been avoided, but the claims were directed to a mental act, since they involved “a significant level of abstraction and generality”: at 532. Note also that the relevance of claims reading on mental steps was acknowledged in the pre-EPC case of *IBM's Application* [1980] FSR 564 at 567 (discussed in Chapter 3 at 102) where it was said “such an operation could in theory be done without the need for any automatic aids but in practice needs to be automatically computed.” Given the strong endorsement of this case in Australian jurisprudence, this arguably represents the position in Australia. This position is further fortified by the exclusion for schemes, rules and plans, which find expression in the same section of Article 52 EPC as the mental steps doctrine, suggesting a close relationship between these areas.

³⁸ *Greenewalt v Stanley Co of America* 12 USPQ 122 (1931) at 123, discussed in Chapter 3 at 78.

³⁹ *In re Abrams* 89 USPQ (BNA) 266 (1951), discussed in Chapter 3 at 78.

⁴⁰ *In re Abrams* 89 USPQ (BNA) 266 (1951), discussed in Chapter 3 at 78.

⁴¹ *In re Abrams* 89 USPQ (BNA) 266 (1951) at 269.

⁴² One possible foundation was put forward in *Prater*, namely inconsistency with a right of freedom of speech, but it was never explored in any detail. The possible relevance of this foundation is explored later in this thesis.

⁴³ *In re Bernhart* 417 F.2d 1395 (1969) at 1401 (Chapter 3 at 82).

⁴⁴ *In re Benson* 441 F.2d 682 (1971) at 687; see Chapter 3 at 84.

⁴⁵ *In re Prater* 415 F.2d 1378 (1968) at 1389; *In re Prater* 415 F.2d 1393 (1969) at 1403. This notion underpins the later developments in *In re Bernhart* 417 F.2d 1395 (1969) and *In re Musgrave* 431 F.2d 882 (1970) (Chapter 3, page 83).

Why is the patenting of software problematic?

to avoid the exclusion. In this sense the approach of the CCPA proceeded along similar lines to the European Patent Office's interpretation of the computer program exclusion wherein the execution of the program on computer hardware was considered to render such objections to its patentability nugatory. These assumptions about the relationship between computer and mind, and the relationship between patent law and the First Amendment⁴⁶ were never given direct consideration. In *Musgrave*⁴⁷ the doctrine was dismissed in favour of a requirement, based on the statutory language in 35 USC 101 that the claimed invention "be in the technological arts".⁴⁸ Further, the Supreme Court in *Gottschalk v Benson* had an opportunity to address the mental steps doctrine, it redirected attention to the patentability of mathematical algorithms.

Given the importance and complexity of the relationship between computers and mental processes,⁴⁹ it is unfortunate that this doctrine was abandoned,⁵⁰ as it forced the consideration of this interplay directly. Such consideration might have directly addressed the admittedly difficult issues of the relationship between the computer and the human mind,⁵¹ and whether it is appropriate that patent law, by "conferring a broad property right in cognitive process steps appropriates publicly necessary information to private use".⁵²

It may be that the diversion of attention to the patentability of algorithms really only amounts to a change in nomenclature.⁵³ However it does seem to suggest a (similarly misunderstood) distinction between mathematics and software development, and has done nothing but obfuscate the issues. This is because it encourages speculation about what is mathematical and what is not, and about the differences between pure and applied mathematics.

Intellectual information exclusion

A similar approach developed in UK law which held mere intellectual information to be non-patentable subject matter.⁵⁴ This same exclusion has been adopted in Australia, although rejections are typically referred to other exclusionary grounds, such as the lack of a practical

⁴⁶ See above n42.

⁴⁷ *In re Musgrave* 431 F.2d 882 (1970).

⁴⁸ *In re Musgrave* 431 F.2d 882 (1970) at 893. The interrelationship between this doctrine and the technological or useful arts limitation will be advanced later in this thesis.

⁴⁹ This interplay is explored in great detail later in this thesis.

⁵⁰ Samuelson notes that with the exception of *In re Meyer* 688 F.2d 789 (1982) and *In re Sarkar* 588 F.2d 1330 (1978) "the 'mental process' rationale for rejecting program-related inventions rarely appears in the case law.": Pamela Samuelson, "Benson Revisited: The Case Against Patent Protection for Algorithms and Other Computer Program-Related Inventions," (1990) 39 *Emory Law Journal* 1025 at 1043, note 59.

⁵¹ "That a human being is a kind of machine and that the human mind is (or is like) a digital computer are among the central theories of cognitive science and of the artificial intelligence disciplines.": Pamela Samuelson, "Benson Revisited: The Case Against Patent Protection for Algorithms and Other Computer Program-Related Inventions," (1990) 39 *Emory Law Journal* 1025 at 1044, note 60. See also *ibid* at 1060, note 122: "[T]he phrases 'mental process' and 'mental steps,' while admittedly awkward and not entirely apt for describing some important patentable subject matter problems, may be closer to the heart of the problem with software patentability than the term 'algorithm,' which in the first post-Benson case law comes to dominate the debate." See also Allen Newell, "Response: The Models Are Broken, The Models Are Broken," (1986) 47 *University of Pittsburgh Law Review* 1023 at 1025.

⁵² Dan L Burk, "Patenting speech" (2000) 79 *Texas Law Review* 99 at 140.

⁵³ "Sequences of mental steps and algorithms are the same thing. Any attempt in the law to make distinctions that depend upon contrasting mental steps versus algorithms is doomed to eventual confusion.": Allen Newell, "Response: The Models Are Broken, The Models Are Broken," (1986) 47 *University of Pittsburgh Law Review* 1023 at 1025.

⁵⁴ See for example *Re Cooper's Application* (1901) 19 RPC 53 (AG), discussed in Chapter 2 at 51. Similarly in *Fishburn's Application* (1940) 57 RPC 245 the arrangement of information on both halves of a ticket so as to allow

Chapter 4: Why Software Patents are a Problem

application,⁵⁵ or as falling within the fine arts.⁵⁶ Such an exclusionary basis also exists in the US, where it is known as the printed matter doctrine.⁵⁷ Although the name suggests a different focus, the substantive issue addressed is the same, namely whether what is being claimed is the intellectual content, as opposed to the mechanical layout of text. On both approaches, the former is non-patentable.

On a theoretical level, the intellectual information exclusion runs into trouble. This is because “[p]atents are essentially about information as to what to make or do.”⁵⁸ Thus the subject matter of patents is intangible, in that although patent claims may describe the features and interrelationship of physical entities, these descriptions are themselves only representative. In other words, the determination of subject matter issues is conducted at an abstract level. Is it any wonder that in these circumstances it may seem sufficient to find that abstract claims bearing a mere representational link to a physical entity might seem sufficient?

This issue is even more pronounced when the subject matter relates to a generally intellectual process of designing and implementing software ideas, which takes as its input an abstract idea, and descends through various levels of abstraction until ultimately arriving at an intangible output.⁵⁹ It is at the various stages of this process, where the distinctions are the slightest, that the most resolute adherence to such distinctions is required.

Theoretical problems aside, applying this exclusion leads to trying to distinguish the content from the presentation of information.⁶⁰ Characterisation of the invention in this context is

it to be torn in half without losing the requisite information was held to be patentable as it served a mechanical purpose). In contrast, in *Pitman's Application* [1969] RPC 646, a method of teaching the pronunciation of language by visual means was held to serve a functional purpose and thus constitute patentable subject matter. In the UK, this exclusion is now statutorily enacted, as required by the European Patent Convention. In *Fujitsu Ltd's Application* [1996] RPC 511 (Chapter 3, at 111) both Hearing Officer Haselden and the Laddie J on appeal rejected program as concerned with information content/intellectual process, although the Court of Appeal (*Fujitsu Ltd's Application* [1997] RPC 608) looked instead to the absence of a technical contribution. It may be that these are but two sides of the same coin.

⁵⁵ *Grant v Commissioner of Patents* (2006) 154 FCR 62 (Chapter 3 at 128) at 65-66: “It has long been accepted that ‘intellectual information’ ... [is] not patentable.” See *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 264, where the court stated that a non-patentable ‘discovery’ would lie where there was only “abstract information without any suggestion of a practical application”.

⁵⁶ It will be recalled from Chapter 2, that the distinction between useful and fine arts distinction was set out in NRDC at 275 (see Chapter 2 at 26). That intellectual information falls within the fine arts was explicitly stated in *Virginia-Carolina Chemical Corp's Application* [1958] RPC 35 at (ref). See also *CCOM Pty Ltd v Jiejing Pty Ltd* (1993) 27 IPR 577 at 594, cited in *CCOM Pty Ltd v Jiejing Pty Ltd* (1994) 51 FCR 260 (Chapter 2, page 123) at 286, where the trial judge noted that “[t]he formulation of ... criteria and their use as rules to organise and process data... are the product of human intellectual activity lying in the fine arts and not the useful arts.”

⁵⁷ “The mere arrangement of printed matter on a sheet or sheets of paper ... does not constitute ‘any new and useful art, machine, manufacture or composition of matter’ [as required by the US Patents Act]”: *In re Russell* (1931) 48 F.2d 1395. As with UK cases discussed in above, where a claimed invention demonstrates a functional purpose, also phrased as ‘mechanical inventiveness’, it will not fall foul of this exclusion. See for example *Flood v Coe* 31 F Supp 348 (1940) at 349 where the “unique relationship between the physical structure and the printed matter” on claims for a price tag meant that it amounted to “more than an arrangement of printed matter on a piece of paper”.

⁵⁸ *Aerotel Ltd v Telco Holdings Ltd and in the Matter of Macrossan's Application* [2006] EWCA Civ 1371; [2007] RPC 7 at [32].

⁵⁹ See Chapter 1, Section 3, in particular the diagram at 26. See also Gary Dukarich, “Patentability of Dedicated information Processors and Infringement Protection of Inventions that Use Them” (1989) 29 *Jurimetrics Journal* 135 at 140-143.

⁶⁰ In *Nelson's Application* [1980] RPC 173 at 176 the question was said to be “whether a claim ... is a claim to a manner of new manufacture or whether it is really directed to monopolising something which is of a purely intellectual nature”.

Why is the patenting of software problematic?

of great importance. An example is *Cooper's Application*,⁶¹ where the former was held non-patentable *inter alia* because it belonged to the realm of copyright law.⁶²

Since the time of *Coopers' Application* however, the exclusion has been interpreted narrowly, dispensing with the requirement of a physical product as enunciated in that case, requiring instead only a "human act of inventiveness and that such act be the source of the subject matter's demonstrated practical utility."⁶³

In the context of computer programs, the exclusion has been addressed by attempting to delineate between functional and informational aspects, by distinguishing code from data. See for example *Badger's Application*⁶⁴ where a distinction was drawn between the "preparation, tabulation and codification of data" which was held to be "conceptual in character, lacking in ... concrete actuality which differentiates substance from notion"⁶⁵ whereas the "conditioning of the computer to permit computation"⁶⁶ (that is, the code) was found to be appropriate subject matter. Similarly *Slee & Harris* the term "data" was held to be equivalent to intellectual information.⁶⁷

However, computer scientists have long understood the difficulty in computer science on the distinction between code and data. A simple example serves to illustrate the fallacy of the distinction. Building on the loyalty program example from the *Catuity* case,⁶⁸ one might assume that the number of points awarded per dollar spent is different in each store. This information might be represented as data in the following table:

Store	Points per dollar
Grocery retailer	2
Hardware chain	3
Airline	1

Table 4.1: Frequent flyer points look-up table

The code to look up the points per dollar, given a particular store name may then look as follows:

```
def points_per_dollar(store_name):  
    return lookup_table(store_name, column=2)
```

⁶¹ *Re Cooper's Application for a Patent* (1901) 19 RPC 53 (AG) (Chapter 2, page 51).

⁶² The relationship between copyright and patent law was explored (albeit from the copyright perspective) in *Baker v Selden* 101 US 99 (1879) in which it was held that the use of "useful art" (a system of book keeping) explained in a book was not copyright infringement, and could only be protected through patent law. See also Dan L Burk, "Patenting speech," (2000) 79 *Texas Law Review* 99 at 142-144.

⁶³ Justine Pila, "Inherent patentability in Anglo-Australian law: a history," (2003) 14 *Australian Intellectual Property Journal* 109 at 162-163.

⁶⁴ *Badger Co Inc's Application* [1969] FSR 474 (Chapter 3 at 99).

⁶⁵ *Badger Co Inc's Application* [1969] FSR 474 at 476.

⁶⁶ *Badger Co Inc's Application* [1969] FSR 474 at 476.

⁶⁷ *Slee & Harris' Application* [1966] FSR 51 at 53 (Chapter 3 at 99).

⁶⁸ *Welcome Real-Time SA v Catuity Inc and Ors* (2001) 113 FCR 110 (Chapter 3 at 125).

Chapter 4: Why Software Patents are a Problem

In these circumstances, the distinction between code and data is easy to define. The data is what is in the table, and the code defines the process by which that data is extracted. However, the same functionality might also be expressed entirely as code (that is, without using a table) by putting it in the following form:

```
def points_per_dollar(store_name):  
    if store_name == 'Grocery retailer':  
        return 2  
    elif store_name == 'Hardware chain':  
        return 3  
    else:  
        return 1.5
```

In this second example, the distinction between code and data is less well defined. The data is embedded within the code, and less distinguishable from the functional aspects which surround it. Further enhancements to the points scheme can be imagined which might blur the distinction even more:

1. where the hardware chain offers double points from 1 January to 14 February each year; and
2. where there is interlinking between stores, where a purchase at both the grocery retailer and airline in the same months gets a 500 point bonus.

Which approach is the best probably depends on the intended purpose, scale, complexity and performance of the software component which is being written.⁶⁹ But the two approaches outlined are functionally equivalent, and the claims which cover them would be identical. So it can be safely asserted from this that an attempt to draw generalised distinctions between code and data is bound to fail.

Another example further illustrates the troublesome nature of the distinction. Look at the following series of numbers:

0, 1, 1, 2, 3, 5, 8, 13, 21, ...

This is the Fibonacci series, where each number is equal to the sum of the two numbers preceding it in the series. Although this may seem a strained example, the Fibonacci series appears in real-world phenomena, and so might be useful in creating computer-based models of these things. One way of storing this series would be as a look up table, just like the loyalty program table above:

⁶⁹ Using such an approach for anything other than a small dataset which is unlikely to require alteration at a later date would be bad software design, but the example illustrates the point nonetheless.

Why is the patenting of software problematic?

Fibonacci number
0
1
1
2
3
...

Table 4.2: Fibonacci series look-up table

This may be a suitable approach where the number of elements of the series to be stored is small. Another approach however, would be to implement code to calculate each row as required. The code to achieve this might be written as follows:⁷⁰

```
def fib(n):
    if n == 0:
        return 0
    elif n == 1:
        return 1
    else:
        return fib(n-1) + fib(n-2)
```

Complex data will generally be the norm in software development, since

[p]rograms are typically designed to model complex phenomena, and more often than not one must construct computational objects that have several parts in order to model real-world phenomena that have several aspects.⁷¹

This complexity itself gives rise to more complex interrelationships of code and data.⁷² Code and data may also be entirely interchangeable labels depending on context. For example, from the perspective of the programmer writing a compiler or interpreter for a particular programming language, programs written in that language are just data to be fed into the compiler.

⁷⁰ It probably wouldn't be written in this form in practice, as the recursive nature of the definition means that to calculate the 100th element of the series in this form "requires 708449696358523830149 function calls, which would take longer than the age of the universe to execute even though the stack depth would never exceed 100": "Fibonacci numbers (Python)" <[http://en.literateprograms.org/Fibonacci_numbers_\(Python\)>](http://en.literateprograms.org/Fibonacci_numbers_(Python)>) (24 August 2010). This resource sets out a number of increasingly efficient methods of calculating members of the series, which demonstrate that this is in fact a proper method of storing such data.

⁷¹ Harold Abelson and Gerald Jay Sussman, "Chapter 2: Building Abstractions with Data" in *Structure and Interpretation of Computer Programs* (2nd ed, 1996) at 13.

⁷² At the simplest end of the complexity scale would be whole numbers. These can become more complex by their combination into compound data types such as lists, series and trees (or hierarchies). A level of abstraction above this is to make numbers representative of other things, such as characters, pixels, colours and so on. Further complexity is possible when compound data types are capable of containing arbitrary data types, wherein the code processing the data would be the same for a list of numbers, a list of Greek letters, or even a list of lists. See Harold Abelson and Gerald Jay Sussman, "Chapter 2: Building Abstractions with Data" in *Structure and Interpretation of Computer Programs* (2nd ed, 1996).

Chapter 4: Why Software Patents are a Problem

Similarly, executable software, or object code, is data which the compiler or interpreter produces.

In fact, this implicit interchangeability lies at heart of theoretical foundation of modern computing. It is inherent in the Turing machine definition that it is powerful enough to treat its own code as data.⁷³

Australian cases directed towards computer related inventions have glossed over the intellectual aspects of computer software in favour of a focus on the end result of its execution.⁷⁴ The problem with focusing on the ends is that it confuses the distinction between hardware and software, by looking to the physical effect of executing software, rather than considering software itself as a standalone entity with separate, although related, existence. Put in the language of the Court in *RCA Photophone*, but contrary to what the Court there urged, there has been a focus on the ends rather than the means.⁷⁵

Thus on the approaches to excluding subject matter just reviewed, namely pre-emption, mental steps and intellectual information, it seems they touch upon the right issues. Where they have failed is in a lack of direct engagement with the underlying issues, instead either jumping directly to conclusions, relying on illusory distinctions, or skating off into tangential inquiries. Meeting the challenge of a theoretical underpinning to abstract subject matter is something to which the latter part of this thesis is directed.

Given the potential problems which highly abstract subject matter creates it becomes clear that considering the context of the claimed invention, to determine the level of abstraction and consequent relationship with the physical aspects of the device, becomes of particular importance.

The term context could perhaps be replaced by physicality, in that the physicality of a claimed invention is an important consideration in determining the nature of what is said to have been invented. Similarly, the notion of context could be subsumed within the discussion of abstraction above, as the absence of physicality might be taken to be confirmation of an alleged invention's abstract nature. But looking at the context in which a claimed invention is to operate is an inquiry preliminary to the determination of whether an invention is abstract or sufficiently limited by physical constraints since it is not until it can be said with confidence what the innovative advance claimed is, that one can move on to look at why or why not it is patentable subject matter.

This context issue is complicated by the need to identify the contribution, or advance which the invention entails. As a result, there is an actively propounded concern that such issues

⁷³ See Piers Cawley, "Code is data, and it always has been" on *Just a Summary* <<http://www.bofh.org.uk/2008/04/07/code-is-data-and-it-always-has-been>> (1 September 2010). Cawley draws this inference from Turing's description of the halting problem, which assumes that it is possible for a Turing machine to treat a program written for it as data which can be fed into a second program. The second program in the halting problem context, is a program which is to detect whether the first program will ever reach a conclusion. Turing's proof establishes that such a program cannot be written, but the ability to treat a program as data for a second program is a necessary precondition of the proof.

⁷⁴ See *CCOM Pty Ltd v Jiejing Pty Ltd* (1994) 51 FCR 260 at 295 where the Full Federal Court interpreted *NRDC* as requiring only "a mode or manner of achieving an end result which is an artificially created state of affairs of utility in the field of economic endeavour". In *Grant v Commissioner of Patents* (2006) 154 FCR 62 at 70, the Court added a further requirement that a "concrete, tangible, physical, or observable effect" be produced.

⁷⁵ *RCA Photophone Ltd v Gaumont-British Picture Corporation* (1936) 53 RPC 167 at 191 (Chapter 2, at 54).

Why is the patenting of software problematic?

intrude upon the other indicia of patentability, namely novelty and inventive step. This interrelationship is not surprising however, given that these indicia originally co-existed with inherent patentability as a single consideration. It was only later in the history of patent law that they gained any independent existence. In fact, for this reason it has been asserted that the software patent problem is in fact an inventive step issue rather than one to be dealt with under the patentable subject matter penumbra. That claim is dealt with in Section 4 on page 181 below.

Despite the importance of characterising the claimed invention, the development of the law in Australia, the US and EU has seen the context issue regularly overlooked. Since software interacts with the computer on which it is run, and with various associated physical devices such as printers, scanners and storage devices, one does not have to look far to find something which renders any claimed software innovation physical enough to satisfy the inquirer that what is claimed is inherently patentable. To properly understand how this mischaracterisation occurs, it is apposite to firstly recap the relevance of physicality to the inherent patentability inquiry.

Physicality

In Chapter 2 it was established that a physical existence is a time-worn feature of patentable subject matter, whether having always existed, or having taken an explicit form somewhere between the time of *Boulton v Bull*⁷⁶ and its expression in Morton J's "vendible product" test.⁷⁷ In Australian law the vendible product test continues to inform the patentable subject matter issue through its expanded re-characterisation in *NRDC*,⁷⁸ in particular the requirement of an artificially created state of affairs.⁷⁹

In early Australian jurisprudence, the High Court in *Maeder v Busch* noted that "the ultimate end in view [of patentable subject matter] is the production or treatment of, or effect upon, some entity".⁸⁰ The modern version of this "useful effect" test, although arguably derived from US law, was first put forward in *IBM v Commissioner of Patents*⁸¹ where Burchett J stated that the difference between discovery and invention is distinguished by reference to "production of some useful effect".⁸² As noted above however, the focus on the end result is problematic. Any reference to an end result should be interpreted narrowly, as it was in *RCA Photophone* to "what

⁷⁶ All four judgments in *Boulton v Bull* (1795) 126 ER 651 (discussed in Chapter 2 at 42) include some form of physicality requirement. Eyre CJ held that a method to be patentable should be "embodied and connected with corporeal substances": at 667; Heath J limited patentability to two physical classes, namely "machinery [and] substances (such as medicines) formed by chemical and other processes": at 660; Buller J noted that a method was not patentable until its inventor had "carried it into effect and produced some new substance": at 663; and physicality is implicit in Rooke J's "new mode of construction": at 659, although it is acknowledged that his Honour did suggest that principles in the abstract might be patentable.

⁷⁷ *Re GEC's Application* (1941) 60 RPC 1 at 4 (Chapter 2 at 56).

⁷⁸ See for example *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 277, and in particular the discussion of the case in Chapter 2 beginning on page 66.

⁷⁹ The link between the two is expressed *R v Wheeler* (1819) 106 ER 392 (KB) at 395 (Chapter 2 at 45) as a requirement of "[s]omething of a corporeal and substantial nature, something that can be made by man from the matters subjected to his art and skill" (emphasis added).

⁸⁰ *Maeder v Busch* (1938) 59 CLR 684 at 705, discussed in Chapter 2 at 20.

⁸¹ *International Business Machines Corporation v Smith, Commissioner of Patents* (1991) 33 FCR 218 (Chapter 3 at 121).

⁸² *International Business Machines Corporation v Smith, Commissioner of Patents* (1991) 33 FCR 218 at 224.

Chapter 4: Why Software Patents are a Problem

practically achieved in physical fact”.⁸³ On this basis, the end result is not the execution of the software on the computer, but software created through the application of the inventive faculty. Whether this has any tangible physical manifestation is an entirely different question.

Perhaps the low point of support for physicality in Australia came in *Catuity* where Heerey J expressed doubt that such a requirement existed,⁸⁴ but in any event was satisfied that it was present along similar lines to the EU “any hardware” approach. The Full Court decision in *Grant* has clearly rejected this position⁸⁵ explaining that what was meant in *NRDC* by an artificially created state of affairs, was not a “useful effect” in the broad sense, but a “useful product”⁸⁶ or a “concrete, tangible, physical, or observable effect”.⁸⁷

Physicality has also long been part of the US notion of patentability, whether accepted as a requirement⁸⁸ or as a “useful and important clue”.⁸⁹ Physicality is impliedly required by even the broadest statements of patentability in that jurisdiction. For example, the US Supreme Court in *Chakrabarty*⁹⁰ that “anything under the sun made by man” is patentable was subsequently relied on for its allegedly expansive interpretation of patentable subject matter. However, at the very least the judgment implies a requirement of corporeality, in that any *thing* which is patentable must be able to exist “under the sun”. Taking this view is further justified by the fact that the Court was in that case discussing only 2 branches of patentable subject matter under the US *Patents Act*, namely machines and manufactures.⁹¹

At the US Federal Circuit level, the *Freeman-Walter-Abele* (FWA) test⁹² was set out in *Abele*⁹³ as requiring that the claimed process be “applied to physical elements or process steps”.⁹⁴ Despite this test being abandoned in *Alappat*,⁹⁵ the physicality of a claimed invention continued to inform the new requirement of a “useful, concrete and tangible result”.⁹⁶ A less stringent approach was however advocated, as it was only necessary that the claims cover something which *represented* measurable physical phenomenon.⁹⁷

Thankfully,⁹⁸ in the wake of *Bilski* this approach is not likely to be supported. Whilst the impact of *Bilski* in the US is still being worked out, an early indication of the way the patentable

⁸³ *RCA Photophone Ltd v Gaumont-British Picture Corporation* (1936) 53 RPC 167 (Chapter 2 at 54).

⁸⁴ *Welcome Real-Time SA v Catuity Inc and Ors* (2001) 113 FCR 110 at 137 (Chapter 3, at 125).

⁸⁵ That such physicality was not necessary was an argument put forward by the appellant. See *Grant v Commissioner of Patents* (2006) 154 FCR 62 (Chapter 3 at 128).

⁸⁶ *Grant v Commissioner of Patents* (2006) 154 FCR 62 at 73 (emphasis added).

⁸⁷ *Grant v Commissioner of Patents* (2006) 154 FCR 62 at 70.

⁸⁸ See for example *Mackay Radio v RCA* 306 US 86 (1939) (Chapter 2 at 59) which required a “novel and useful structure” for patentability.

⁸⁹ *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3258 per Breyer J.

⁹⁰ *Diamond v Chakrabarty* 447 US 309 (1980) (Chapter 3 at 90).

⁹¹ See *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3249 per Stevens J. His Honour notes that the quoted statement could not be taken to extend to processes in any event, as a process could not “be comfortably described as something ‘made by man’”.

⁹² It will be recalled that the *Freeman-Walter-Abele* test is short-hand for the approach suggested by the Federal Circuit in a series of three cases: *In re Freeman* 573 F.2d 1243 (1978) (Chapter 3 at 89); *In re Walter* 618 F.2d 766 (1980) (Chapter 3 at 89) and *In re Abele* 684 F.2d 902 (1982) (Chapter 3 at 93).

⁹³ *In re Abele* 684 F.2d 902 (1982).

⁹⁴ *In re Abele* 684 F.2d 902 (1982) at 907.

⁹⁵ *In re Alappat* 33 F.3d 1526 (1994) (Chapter 3 at 93).

⁹⁶ *In re Alappat* 33 F.3d 1526 (1994) at 1544.

⁹⁷ See also *State Street Bank & Trust Co v Signature Financial Group Inc* (1998) 149 F.3d 1368 (Chapter 3 at 94) where the “transformation of data, representing discrete dollar amounts” could amount to a useful effect.

⁹⁸ See Robert Plotkin, “Computer Programming and the Automation of Invention,” (2003) 7(2) *UCLA Journal of Law*

Why is the patenting of software problematic?

subject matter inquiry is likely to proceed can be found in the USPTO's recently published Interim Guidance.⁹⁹ The Guidance notes the continued importance of both the machine-or-transformation test, and the physicality of claims:

To date, no court, presented with a subject matter eligibility issue, has ever ruled that a method claim that lacked a machine or a transformation was patent-eligible. However, *Bilski* held open the possibility that some claims that do not meet the machine-or-transformation test might nevertheless be patent-eligible.

Prior to adoption of the machine-or-transformation test, the Office had used the "abstract idea" exception in cases where a claimed "method" did not sufficiently recite a physical instantiation. ... Following *Bilski*, such an approach remains proper.¹⁰⁰

The Interim Guidance also mandates a set of "factors that are to be considered when evaluating patent-eligibility of method claims",¹⁰¹ such factors being derived from "the machine-or-transformation test, which remains a useful investigative tool, and inquiries gleaned from Supreme Court precedent."¹⁰² The Interim Guidance notes that these factors are not exhaustive, merely "useful examples and are not intended to be exclusive or limiting. It is recognized that new factors may be developed, particularly for emerging technologies."¹⁰³ The factors can be summarised as follows:

A. Whether the method involves or is executed by a particular machine or apparatus. If so, the claims are less likely to be drawn to an abstract idea; if not, they are more likely to be so drawn. ...

B. Whether performance of the claimed method results in or otherwise involves a transformation of a particular article. If such a transformation exists, the claims are less likely to be drawn to an abstract idea; if not, they are more likely to be so drawn. ...

C. Whether performance of the claimed method involves an application of a law of nature, even in the absence of a particular machine, apparatus, or transformation. If such an application exists, the claims are less likely to be drawn to an abstract idea; if not, they are more likely to be so drawn. ...

D. Whether a general concept (which could also be recognized in such terms as a principle, theory, plan or scheme) is involved in executing the steps of the method.

and Technology <http://www.lawtechjournal.com/articles/2003/07_040127_plotkin.php> (19 April 2007): "The fact that a particular logical entity may represent a physical quantity or physical entity does not mean that the logical entity is itself physical."

⁹⁹ United States Patent and Trademark Office, "Interim Guidance for Determining Subject Matter Eligibility for Process Claims in View of *Bilski v Kappos*" (2010) 75(143) *Federal Register* 43922.

¹⁰⁰ United States Patent and Trademark Office "Interim Guidance for Determining Subject Matter Eligibility for Process Claims in View of *Bilski v Kappos*" (2010) 75(143) *Federal Register* 43922 at 43924.

¹⁰¹ United States Patent and Trademark Office "Interim Guidance for Determining Subject Matter Eligibility for Process Claims in View of *Bilski v Kappos*" (2010) 75(143) *Federal Register* 43922 at 43924.

¹⁰² United States Patent and Trademark Office "Interim Guidance for Determining Subject Matter Eligibility for Process Claims in View of *Bilski v Kappos*" (2010) 75(143) *Federal Register* 43922 at 43924.

¹⁰³ United States Patent and Trademark Office "Interim Guidance for Determining Subject Matter Eligibility for Process Claims in View of *Bilski v Kappos*" (2010) 75(143) *Federal Register* 43922 at 43924.

Chapter 4: Why Software Patents are a Problem

The presence of such a general concept can be a clue that the claim is drawn to an abstract idea.¹⁰⁴

Thus it seems that although not the sole test, the machine-or-transformation test will continue to dominate the test for patentable subject matter. It is further submitted that the application of a law of nature also involves the concept of physicality, and the presence of a general concept indicates a lack of physicality. So physicality continues to be an important concept in US patentable subject matter doctrine.

It is in EPO jurisprudence where the physicality of claims is given the most lax interpretation. Although the German interpretation of technical character has led to the enunciation of a requirement that the claimed invention demonstrate some causal involvement with physical forces,¹⁰⁵ a different tack has been taken at the EPO. For example, in *Vicom*¹⁰⁶ the EPO held that an alleged invention would be impermissibly “abstract so long as it is not specified what physical entity is *represented*”.¹⁰⁷ This suggests an *Alappat*-style approach to physicality.¹⁰⁸ However recent EPO case law goes further than that, advocating an “any hardware” approach according to which the recitation of a physical feature in the claims will be sufficient to clear the patentability hurdle.¹⁰⁹ This is of course the point at which the approach adopted by the UK courts differs from the EPO, where the recitation of hardware in the claims may well be ignored as a “confusing irrelevance”.¹¹⁰

Mischaracterisation

Even assuming that physicality is an express requirement, this does not dispose of the issue of whether a software-based invention ought to be considered patentable. As the survey of EPO jurisprudence in Chapter 3 suggests, such a requirement can rapidly devolve into a mere form requirement. The cases thus far discussed show that even the strictest physicality requirement is easily satisfied by the following arguments.

Software creates a new machine In the first significant statement on the topic of software patents anywhere in the world, the United States President’s Commission of 1966 report noted, after a substantial period of consultation and consideration, that

¹⁰⁴ United States Patent and Trademark Office “Interim Guidance for Determining Subject Matter Eligibility for Process Claims in View of *Bilski v Kappos*” (2010) 75(143) *Federal Register* 43922 at 43925-43926.

¹⁰⁵ German patent law requires a “plan-conformant activity of using controllable natural forces to achieve a causally overseeable success which is, without mediation by human reason, the immediate result of controllable natural forces”: Bundesgerichtshof (German Federal Court of Justice) 22 June 1976, XZB 23/74 “Dispositionsprogramm”. A partial English translation is available at <<http://swpat.ffii.org/vreji/papri/bgh-dispo76/>> (13 April 2007).

¹⁰⁶ *Vicom/Computer-Related Invention* T208/84 [1987] EPOR 74 (Chapter 3 at 105).

¹⁰⁷ *Vicom/Computer-Related Invention* T208/84 [1987] EPOR 74 at 79.

¹⁰⁸ *Fujitsu Ltd’s Application* [1996] RPC 511 at 519 (Chapter 3 at 111) per Hearing Officer Haselden, who found that the representation of a technical artefact was insufficient to amount to a technical advance.

¹⁰⁹ For example see T258/03 *Hitachi/Auction Method* [2004] EPOR 55 (Chapter 3 at 116) at [4.5] where technical character could be implied by “the physical features of an entity”, which might even be satisfied by “activities which are so familiar that their technical character tends to be overlooked, such as the act of writing using pen and paper”: at [4.6].

¹¹⁰ *Gale’s Patent Application* [1991] RPC 305 at 326.

Why is the patenting of software problematic?

[i]ndirect attempts to obtain patents and avoid the rejection, by drafting claims as a process, or a machine or components thereof programmed in a given manner, rather than as a program itself, have confused the issue further and should not be permitted.¹¹¹

Despite this, courts and tribunals in the US, UK and Australia, when reviewing claims to computer programs, have found sufficient physicality for such methods in the execution of that software on computer. In the US, only three years after the President's Commission, the CCPA in *Prater* held the mental steps doctrine was inapplicable to software claims because the claims included a "disclosure of apparatus for performing the process wholly without human intervention".¹¹² This was further explained in *Bernhart* as follows:

[I]f a machine is programmed in a certain new and unobvious way, it is physically different from the machine without that program; its memory elements are differently arranged. The fact that these physical changes are invisible to the eye should not tempt us to conclude that the machine has not been changed. If a new machine has not been invented, certainly a 'new and useful improvement' of the unprogrammed machine has been...¹¹³

Similarly in *Musgrave*, the CCPA relied on the performance of the claimed steps "by the disclosed apparatus"¹¹⁴ in holding that the mental steps exclusion did not apply. This interpretation was returned to in the wake of *Benson*, *Flook* and *Diehr* by the Federal Circuit in *Alappat* wherein the Court held that

programming creates a new machine, because a general purpose computer becomes a special purpose computer once it is programmed to perform particular functions pursuant to instructions from program software¹¹⁵

This interpretation was relied on in *State Street Bank v Signature Financial Group*¹¹⁶ to hold that "the transformation of data ... by a machine ... constitutes a practical application of a mathematical algorithm ..."¹¹⁷ and was thus not a non-patentable abstract idea.

In a string of cases before the EPC came into effect in the UK, the Patents Appeal Tribunal relied on just such a characterisation to find software patentable. In *Slee & Harris* the Tribunal held that a computer when programmed amounted to "a machine which is temporarily modified".¹¹⁸ In *Badger's Application* a contrast was drawn between the conceptual and non-patentable preparation of data on the one hand, and on the other the patentable "conditioning

¹¹¹ Presidential Commission on the Patent System, *To Promote the Progress of Useful Arts in an Age of Exploding Technology* (1966) (Chapter 3 at 79).

¹¹² *In re Prater* 415 F.2d 1378 (1968) at 1403 (Chapter 3 at 81).

¹¹³ *In re Bernhart* 417 F.2d 1395 (1969) at 1400 (Chapter 3 at 82).

¹¹⁴ *In re Musgrave* 431 F.2d 882 (1970).

¹¹⁵ *In re Alappat* 33 F.3d 1526 (1994) at 1545 (Chapter 3 at 93).

¹¹⁶ *State Street Bank & Trust Co v Signature Financial Group Inc* 149 F.3d 1368 (1998) (Chapter 3 at 94).

¹¹⁷ *State Street Bank & Trust Co v Signature Financial Group Inc* 149 F.3d 1368 (1998) at 1373.

¹¹⁸ *Slee & Harris' Application* [1966] FSR 51 at 55 (Chapter 3 at 99).

Chapter 4: Why Software Patents are a Problem

of the computing apparatus so as to control its operation, in consonance with corrections laid down by the data record requirements".¹¹⁹ The same Tribunal in *Gever's Application* characterised the claims therein as "data processing apparatus ... constrained to work in a certain way by punched cards which are inserted into it".¹²⁰ On a slightly different tack, it was held in *IBM's Application* that although claims to no more than a standard computer could not be patentable subject matter, "the claim on its true construction ... only covers a computer when it is programmed to produce the required result and does not cover a standard computer as such".¹²¹ This line of cases was relied on in the Australian case of *IBM v Commissioner of Patents*¹²² to support the proposition that the claims to a method of generating a curve on a computer "involve[d] steps which are foreign to the normal use of computers".¹²³ It was further stated, as had been suggested in *Burroughs*,¹²⁴ that such a position was consistent with *NRDC*.¹²⁵

European jurisprudence initially rejected the "new machine" approach. For example, in the *IBM Twins* case¹²⁶ the EPO seemed to acknowledge that this alone is insufficient, holding it necessary that the computer program brings about "a technical effect which goes beyond the 'normal' physical interactions between the program (software) and the computer (hardware) on which it is run".¹²⁷ Requiring something beyond these 'normal' interactions is consistent with the underlying question which the technical contribution requires to be answered, namely, what has the applicant invented?¹²⁸ Whether this includes some aspect of hardware is dependent entirely on the particular claims in question.

However more recent EPO case law has implicitly embraced this approach, with the Board in *Pension Benefits* noting "a computer system suitably programmed for use in a particular field ... has the character of concrete apparatus in the sense of a physical entity, man-made for a

¹¹⁹ *Badger's Application* [1969] FSR 474 at 476 (Chapter 3 at 99).

¹²⁰ *Gever's Application* [1969] FSR 480 at 486 (Chapter 3 at 100). See also *Burroughs Corporation's Application* [1973] FSR 439 at 449 (Chapter 3 at 101) "The programme in fact constrains the apparatus to function in a particular way as long as the apparatus embodies that programme".

¹²¹ *International Business Machines Corporation's Application* [1980] FSR 564 at 568 (Chapter 3 at 102).

¹²² *International Business Machines Corporation v Commissioner of Patents* (1991) 33 FCR 218 (Chapter 3, at 121).

¹²³ *International Business Machines Corporation v Commissioner of Patents* (1991) 33 FCR 218 at 226.

¹²⁴ *Burroughs Corporation's Application* [1973] FSR 439.

¹²⁵ In *Burroughs Corporation's Application* [1973] FSR 439 at 449 it was held that where the method results "in some improved or modified apparatus, or an old apparatus operating in a novel way, with consequent economic importance or advantages in the field of the useful as opposed to the fine arts".

¹²⁶ T1173/97 *IBM/Computer program product* [1999] OJ EPO 609 and T935/97 *IBM's Application* [1999] RPC 861 (Chapter 3 at 114).

¹²⁷ T1173/97 *IBM/Computer program product* [1999] OJ EPO 609 at [13].

¹²⁸ See for example *Diamond, Commissioner of Patents and Trademarks v Diehr et al* (1981) 450 U.S. 175 at 193-194: "The starting point in the proper adjudication of patent litigation is an understanding of what the inventor claims to have discovered"; *In re Grams* 888 F.2d 839 (1989) at 839: "In all instances, this critical question must be answered: 'What did applicants invent?'; *In re Abele* 684 F.2d 902 (1982) at 907 "The goal is to answer the question 'What did applicants invent?'. The court then cited the following approach, as suggested in *In re Sarkar* 588 F.2d 1330 (1978) at 1333: "each invention must be evaluated as claimed: yet semantogenic considerations preclude a determination based solely on words appearing in the claims. In the final analysis under §101, the claimed invention, as a whole, must be evaluated for what it is." In the UK, the approach is the same. According to the Court in *Aerotel Ltd v Telco Holdings Ltd and in the matter of Macrossan's Application* [2006] EWCA Civ 1371; [2007] RPC 7, the first step in determining the subject matter issue is to "properly construe the claim": at [40]. Their Honours elaborated on this as follows. "What has the inventor really added to human knowledge perhaps best sums up the exercise. The formulation involves looking at substance not form – which is surely what the legislator intended.": at [46].

Why is the patenting of software problematic?

utilitarian purpose and is thus an invention. . . .¹²⁹ Similarly, the Board held in *Hitachi* that “the presence of technical character . . . may be implied by the physical features of an entity”¹³⁰ Finally, in *Microsoft* the Board held that “[t]he claimed steps . . . provide a general purpose computer with a further functionality. . . . Moreover, the computer-executable instructions have the potential of achieving [a] further technical effect of enhancing the internal operation of the computer.”¹³¹

The new machine view of software might be, as Samuelson suggests, one which “has some merit as a matter of computer science”.¹³² However, it is wrong to focus on it in the context of the subject matter issue. One reason for this is that “the source program remains at all times outside the machine and separate from it”.¹³³ Thus a modified computer is the *end result* of creating a program, (optionally) compiling it, loading it into a computer and running it. It may even be a *useful* result. But as was suggested in *RCA Photophone* “[t]hat involves an incorrect use of the word “result”, . . . [which is] properly used in this connection as referring, *not to the end, but to the means.*”¹³⁴ In other words, what the applicant has invented may be causally related to the execution of software running on a general purpose of the computer, but this execution is not what the applicant has invented. The relevant means by which this result is achieved is a logical algorithm or procedure for processing data, which may or may not be meaningfully limited in any sense by the hardware of the computer.¹³⁵ Other similarly inappropriate approaches include reliance on the print-out of a receipt as part of the business method in *Catuity*,¹³⁶ or the display of a curve on the screen as part of a claimed curve algorithm in *IBM v Commissioner of Patents*.¹³⁷ To use the words of the US Supreme Court, it is submitted that these amount to nothing more than insignificant post-solution activities.¹³⁸

It should also be noted that some commentators have sought to overcome such criticisms by defining software to cover only an “executable computer program”.¹³⁹ This view might lend more support to the “new machine” argument, in that an executable computer program is

¹²⁹ T931/95 *PBS Partnership/Controlling pension benefits systems* [2002] EPOR 522 at 530 (Chapter 3 at 115).

¹³⁰ T258/03 *Hitachi/Auction Method* [2004] EPOR 55 at [4.5] (Chapter 3 at 116).

¹³¹ T424/03 *Microsoft/Clipboard formats I* [2006] EPOR 39 at [5.2] (Chapter 3 at 117).

¹³² Pamela Samuelson, “Benson Revisited: The Case Against Patent Protection for Algorithms and Other Computer Program-Related Inventions,” (1990) 39 *Emory Law Journal* 1025 at 1045, n 63.

¹³³ Michael C Gemignani, “Legal Protection for Computer Software: The View From ‘79” (1979-1980) 7 *Rutgers Journal of Computers Technology and Law* 269 at 279.

¹³⁴ *RCA Photophone Ltd v Gaumont-British Picture Corporation* (1936) 53 RPC 167 at 191, emphasis added (Chapter 2, 54).

¹³⁵ The requirement of a meaningful limitation is an aspect of the machine-or-transformation test. See *In re Bilski* (2008) 545 F.3d 943 at 962: “the use of a specific machine or transformation of an article must impose meaningful limits on the claim’s scope to impart patent-eligibility.” (citing in support *Gottschalk, Acting Commissioner of Patents v Benson et al* (1973) 409 U.S. 63 at 71-72).

¹³⁶ *Welcome Real-Time SA v Catuity Inc and Ors* (2001) 113 FCR 110.

¹³⁷ *International Business Machines Corporation v Smith, Commissioner of Patents* (1991) 33 FCR 218.

¹³⁸ See *Parker v Flook* (1978) 437 U.S. 584 at 590: “The notion that post-solution activity, no matter how conventional or obvious in itself, can transform an unpatentable principle into a patentable process exalts form over substance. A competent draftsman could attach some form of post-solution activity to almost any mathematical formula; the Pythagorean theorem would not have been patentable, or partially patentable, because a patent application contained a final step indicating that the formula, when solved, could be usefully applied to existing surveying techniques. n11 The concept of patentable subject matter under § 101 is not “like a nose of wax which may be turned and twisted in any direction....” *White v. Dunbar*, 119 U.S. 47, 51.”

¹³⁹ Robert Plotkin, “Computer Programming and the Automation of Invention,” (2003) 7(2) *UCLA Journal of Law and Technology*. A similar approach is advocated in Mark A Lemley & Eugene Volokh, “Freedom of Speech and Injunctions in Intellectual Property Cases” (1998) 48 *Duke Law Journal* 147 at 236.

Chapter 4: Why Software Patents are a Problem

causally close to the new machine. But it does not overcome the criticisms above, since the software is still, even on this definition, independent of the machine. It also glosses over the difficulties which arise in applying the distinction to interpreted languages where source code effectively “executed”.¹⁴⁰ Burk also rightly notes that in any event source code implementations would still face problems with the doctrine of equivalents and contributory infringement.¹⁴¹

Finally, problems with the new machine theory of patentability are reinforced when one considers that a differently configured machine is not the inevitable (or at least only) result of writing a program. Gemignani describes other alternatives as follows:

Just as a person who understands the [programming] language could use a program to reconstruct its algorithm, a person who has an adequate background in electrical engineering could use the program to build an electronic circuit which would carry out the program in conjunction with input and output devices. The program, to a suitable trained engineer, could be used as a circuit diagram or the blueprint for building special purpose hardware. Even if it is not part of a machine, the program could be used to construct a machine of specific design, that design being contained implicitly in the program itself. The program thus stands midway between the abstract solution to a problem and a machine which actually carries out that solution.¹⁴²

Software as a structure on a carrier Another way of reading in physicality into software claims has been by reference to the storage media on which the software is saved, compiled, transferred or loaded, be it hard drive, CD-ROM, USB stick or other device. According to current practice at the United States Patent and Trademark Office:¹⁴³

When functional descriptive material is recorded on some computer-readable medium, *it becomes structurally and functionally interrelated to the medium* and will be statutory in most cases since use of technology permits the function of the descriptive material to be realized.¹⁴⁴

¹⁴⁰ See the discussion of interpreters and interpreted languages in Chapter 1 at 14.

¹⁴¹ Dan L Burk, “Patenting speech,” (2000) 79 *Texas Law Review* 99 at 148-150.

¹⁴² Michael C Gemignani, “Legal Protection for Computer Software: The View From ‘79” (1979-1980) 7 *Rutgers Journal of Computers Technology and Law* 269 at 279.

¹⁴³ United States Patent Office, “2106.01 Computer-related Nonstatutory Subject Matter” in Manual of Patent Examining Procedure (8th Ed, 2001), last revision July 2010 (“MPEP 2106.01”).

¹⁴⁴ MPEP 2106.01 (emphasis added). The MPEP draws a distinction between functional and non-functional descriptive material, which might broadly be considered to be a distinction between code and data, although the MPEP puts it as follows:

“In this context, ‘functional descriptive material’ consists of data structures and computer programs which impart functionality when employed as a computer component. (The definition of “data structure” is ‘a physical or logical relationship among data elements, designed to support specific data manipulation functions.’ The New IEEE Standard Dictionary of Electrical and Electronics Terms 308 (5th ed. 1993).) “Nonfunctional descriptive material” includes but is not limited to music, literary works, and a compilation or mere arrangement of data.”

Why is the patenting of software problematic?

This indicates that so-called *Beauregard* claims,¹⁴⁵ being claims to “computer programs embodied in a tangible medium, such as floppy diskettes”¹⁴⁶ are considered patentable in the US. Also covered are *Lowry* claims,¹⁴⁷ where what is claimed is a novel data structure on a carrier.¹⁴⁸ A less-tangible variation on these types of claims are directed to claims for the propagation of a signal through an intangible medium, however since the issuance of the USPTO interim guidelines in 2005,¹⁴⁹ and the *Nuijten* case,¹⁵⁰ it seems that such claims are no longer considered patentable on the basis that they do not fall within any of the four statutory classes of §101. Early indications after the adoption of the “machine or transformation test” by the Federal Circuit in *In re Bilski* suggested that *Beauregard* and *Lowry* claims remained patentable.¹⁵¹ However, with the recent decision of *CyberSource Corporation v Retail Decisions Inc.*,¹⁵² the (mis)use of this type of claim may be at an end. The court in *CyberSource*, considering the patent-eligibility of “a method for detecting credit card fraud”¹⁵³ looked beyond the form of the claims,¹⁵⁴ and found that what was claimed was in fact the same “abstract mental process”¹⁵⁵ the subject of an earlier claim.

So what falls outside such functional descriptive material on a carrier? According to the USPTO Guidelines (before *CyberSource*), data structures and ‘computer listings’:

¹⁴⁵ *In re Beauregard* 53 F.3d 1583 (1995). The Federal Circuit did not actually make a ruling in *Beauregard*, since the Commissioner for Patents conceded that such claims were statutory, and that the printed matter doctrine was not applicable. Despite this seemingly shaky foundation, the patentability of such claims was thereafter generally accepted. See MPEP 2105.01 at section I.

¹⁴⁶ *In re Beauregard* 53 F.3d 1583 (1995) at 1584.

¹⁴⁷ *In re Lowry* 32 F.3d 1579 (1994).

¹⁴⁸ On first glance such claims are objectionable since the claims seem to be distinguished on the basis of their intellectual content, and run counter to the printed matter doctrine. However, as is the situation at the EPO, the printed matter doctrine bites back at the obviousness stage of the analysis, where “USPTO personnel need not give patentable weight to printed matter absent a new and unobvious functional relationship between the printed matter and the substrate” See MPEP 2106.01, citing in support *In re Lowry* 32 F.3d 1579 (1994) at 1583-1584.

¹⁴⁹ United States Patent and Trademark Office, “Interim Guidelines for Examination of Patent Applications for Patent Subject Matter Eligibility” *Official Gazette Notices*, 22 November 2005, <<http://www.uspto.gov/web/offices/com/sol/og/2005/week47/patgupa.htm>> (24 August 2010).

¹⁵⁰ *In re Nuijten* 500 F.3d 1346 (2007) at 1352 wherein a 2-1 majority of the Court of Appeals for the Federal Circuit held that “transitory electrical and electromagnetic signals propagating through some medium, such as wires, air, or a vacuum ... are not encompassed by any of the four enumerated statutory categories” and was therefore not patentable subject matter.

¹⁵¹ See *Ex parte Bo Li*, Appeal 2008-1213 (BPAI, 2008) wherein the Board of Appeals stated:

It has been the practice for a number of years that a ‘Beauregard Claim’ of this nature be considered statutory at the USPTO as a product claim. (MPEP 2105.01, I). Though not finally adjudicated, this practice is not inconsistent with *In re Nuijten* [(2007) 500 F.3d 1346] (Ibid.). Further, the instant claim presents a number of software components, such as the claimed logic processing module, configuration file processing module, data organization module, and data display organization module, that are embodied upon a computer readable medium. This combination has been found statutory under the teachings of *In re Lowry*, 32 F.3d 1579 (Fed. Cir., 1994). In view of the totality of these precedents, we decline to support the [non-statutory subject matter] rejection under 35 U.S.C. § 101.

¹⁵² Appeal No 2009-1358 (Fed Cir, 2011).

¹⁵³ *CyberSource Corporation v Retail Decisions Inc* Appeal No 2009-1358 (Fed Cir, 2011) at 17.

¹⁵⁴ “Regardless of what statutory category (“process, machine, manufacture, or composition of matter,” 35 U.S.C. § 101) a claim’s language is crafted to literally invoke, we look to the underlying invention for patent-eligibility purposes.”: *CyberSource Corporation v Retail Decisions Inc* Appeal No 2009-1358 (Fed Cir, 2011) at 17.

¹⁵⁵ “[W]e find that claim 3 of the ‘154 patent fails to recite patent-eligible subject matter because it is drawn to an unpatentable mental process—a subcategory of unpatentable abstract ideas.”: *CyberSource Corporation v Retail Decisions Inc* Appeal No 2009-1358 (Fed Cir, 2011) at 9.

Chapter 4: Why Software Patents are a Problem

Data structures not claimed as embodied in computer-readable media are descriptive material per se and are not statutory because they are not capable of causing functional change in the computer.

...

Similarly, computer programs claimed as computer listings per se, i.e., the descriptions or expressions of the programs, are not physical “things.” They are neither computer components nor statutory processes, as they are not “acts” being performed. Such claimed computer programs do not define any structural and functional interrelationships between the computer program and other claimed elements of a computer which permit the computer program’s functionality to be realized.¹⁵⁶

An equivalent approach has also been adopted in recent EPO jurisprudence in *Hitachi*¹⁵⁷ and *Microsoft*.¹⁵⁸

Such claims are highly desirable from a litigation perspective since they do “not require any method steps to be performed to show infringement. Instead, direct infringement can be shown through the storage of the claimed data structure [or computer program] in a computer-readable medium”.¹⁵⁹ On this approach, the “claim format is particularly helpful in proving that manufacturers and sellers of computer-readable media ... are direct infringers.”¹⁶⁰

The main problem with characterising software as a product on a carrier was aptly identified by the hearing officer in *Texas Instruments’ Application*,¹⁶¹ who noted that the carrier is a “record of the programme and not the programme itself”. A similar approach was adopted in *Gale*.¹⁶² At first instance Aldous J held that “a ROM is *more than a carrier*, it is a manufactured article having circuit connections which enables to program to be operated.”¹⁶³ On appeal their Lordships held that the ROM was “merely the vehicle used for carrying [a series of instructions].”¹⁶⁴ As such it was “it is convenient and right to strip away, as a confusing irrelevance, the fact that the claim is for ‘hardware’.”¹⁶⁵

Hardware-software equivalence From a purely functional perspective it is true that “a software process is often interchangeable with a hardware circuit”.¹⁶⁶ It has also been argued that

¹⁵⁶ MPEP 2106.01.

¹⁵⁷ T258/03 *Hitachi/Auction Method* [2004] EPOR 55.

¹⁵⁸ T424/03 *Microsoft/Clipboard formats I* [2006] EPOR 39.

¹⁵⁹ Daniel W McDonald, Robert A Kalinsky and William D Schulz, “Software Patent Litigation” ABA Section of Litigation, Intellectual Property Litigation Committee Roundtable Discussion, April 2006 <<http://euro.ecom.cmu.edu/program/law/08-732/Patents/SoftwarePatentLitigation.pdf>> (20 August 2010) at 3.

¹⁶⁰ Daniel W. McDonald, Robert A. Kalinsky and William D. Schulz, “Software Patent Litigation” ABA Section of Litigation, Intellectual Property Litigation Committee Roundtable Discussion, April 2006 <<http://euro.ecom.cmu.edu/program/law/08-732/Patents/SoftwarePatentLitigation.pdf>> (20 August 2010) at 3.

¹⁶¹ *Texas Instruments Inc’s Application* (1968) 38 AOJP 2846 (Chapter 3 at 119).

¹⁶² *Gale’s Patent Application* [1991] RPC 305 (Chapter 3 at 109)

¹⁶³ *Gale’s Patent Application* [1991] RPC 305 at 316-317.

¹⁶⁴ *Gale’s Patent Application* [1991] RPC 305 at 325 per Nicholls LJ.

¹⁶⁵ *Gale’s Patent Application* [1991] RPC 305 at 326.

¹⁶⁶ David S Bir, “The Patentability Of Computer Software After Alappat: Celebrated Transformation or Status Quo?” (1995) 41 *Wayne L Rev* 1531 at 1551, fn 121. See also Pamela Samuelson et al, “A Manifesto Concerning the

Why is the patenting of software problematic?

“[a] program’s essential characteristic is the technical functionality represented in the instructions for the computer, and in this way is no different to specialised hardware”.¹⁶⁷ As a result, it is tempting to conclude that “the software invention is no less tangible, useful, or worthy of patent protection.”¹⁶⁸ Such a conclusion has often been espoused in case law.

In *Vicom*¹⁶⁹ the EPO stated that “the choice between [hardware and software] is not of an essential nature but is based on technical and economic considerations which bear no relationship to the inventive concept as such”.¹⁷⁰ The Board’s reasons also regularly emphasise this interchangeability.¹⁷¹ Browne-Wilkinson VC in *Gale’s Application*¹⁷² made a similar assumption when he noted that “difficult cases can arise where *the computer program, whether in hardware or software, produces a novel technical effect*”¹⁷³

In the first instance decision of *Fujitsu*,¹⁷⁴ his Honour stated a strong view that the two were entirely equivalent:

In accordance with these principles, just as it would be possible to obtain a patent, considerations of novelty aside, for a faster chip or a more effective storage medium or a computer containing such a chip or storage medium, there is no reason in principle or logic why modification of the computer to achieve the same speed or storage increase by means of software should be excluded from protection. The fact that the advance is achieved in software rather than hardware should not affect patentability. To use in a slightly different context Nicholls L.J.’s words from *Gale’s Application*, that would be to exalt form over substance. Similarly if a new process achieved by mechanical means would be patentable, there is no reason why the same process achieved by computer means should be any less patentable. If that is so, it does not matter whether the patent claims are drafted in terms of a process controlled by a computer, a computer when programmed in a particular way or a method of controlling a computer. In each case the substance of the invention is the same.¹⁷⁵

In the US, a similar view was espoused by Rich J in *Alappat*:¹⁷⁶

Legal Protection of Computer Programs” (1994) 94 *Columbia Law Review* 2308 at 2319.

¹⁶⁷ Advisory Council on Intellectual Property, *Report on a Review of the Patenting of Business Systems* (2003) <<http://www.acip.gov.au/library/bsreport.pdf>> (accessed 19 April 2007) at 8.4.1, summarising the arguments of IBM put forward at the European Patent Office International Academy Forum on the Protection of Computer-related and Business Model Inventions in November 1992.

¹⁶⁸ Joseph R Brown, “Software Patent Dynamics: Software As Patentable Subject Matter After State Street & Trust Co.,” (2000) 25 *Oklahoma City University Law Review* 639 at 660.

¹⁶⁹ T208/84 *Vicom/Computer-Related Invention* [1987] EPOR 74 (Chapter 3 at 105).

¹⁷⁰ *Vicom/Computer-Related Invention* T208/84 [1987] EPOR 74 at 80.

¹⁷¹ See *Vicom/Computer-Related Invention* T208/84 [1987] EPOR 74 at 79: “The technical means might include a computer comprising suitable hardware or an appropriately programmed general purpose computer”; *Ibid* at 80: “a program (be this implemented in hardware or in software)” and also referring to “a specified program (whether by means of hardware or software)”.

¹⁷² *Gale’s Patent Application* [1991] RPC 305 (Chapter 3 at 109).

¹⁷³ *Gale’s Patent Application* [1991] RPC 305 at 333 (emphasis added).

¹⁷⁴ *Fujitsu Ltd’s Application* [1996] RPC 511 (Chapter 3 at 111).

¹⁷⁵ *Fujitsu Ltd’s Application* [1996] RPC 511 at 531 per Laddie J. This point was not considered by the Court of Appeal in when overturning his Honour’s decision: *Fujitsu Ltd’s Application* [1997] RPC 60.

¹⁷⁶ *In re Alappat* 33 F.3d 1526 (1994) (Chapter 3 at 93).

Chapter 4: Why Software Patents are a Problem

Inventors may express their inventions in any manner they see fit, including mathematical symbols and algorithms. Whether an inventor calls the invention a machine or a process is not nearly as important as the invention itself. Thus, the inventor can describe the invention in terms of a dedicated circuit or a process that emulates that circuit. Indeed, the line of demarcation between a dedicated circuit and a computer algorithm accomplishing the identical task is frequently blurred and is becoming increasingly so as the technology develops. In this field, *a software process is often interchangeable with a hardware circuit. Thus, the Board's insistence on reconstruing Alappat's machine claims as processes is misguided when the technology recognizes no difference and the Patent Act treats both as patentable subject matter.*¹⁷⁷

However, it is submitted that the physicality of an invention is a key consideration in patent law. As the review of the history of patentable subject matter in Chapter 2 should make clear, physicality is traditionally a requirement of patent law. Despite the importance often put upon the flexibility to adapt patent law to new conceptions of technology, it is submitted that this physicality requirement ought not be abandoned. The distinction between physical and intangible subject matter is considered in greater detail in Chapter 6, but it is sufficient to say that in summary, the physicality of an invention will have impacts both on the commercialisation of the product, the structure of the market, and most importantly, the nature of the creative process by which such products are created.

A similar criticism can be made of the claim that an extension of patents to software in *IBM v Commissioner of Patents*¹⁷⁸ could be grounded in an *analogy* between the chemical process of NRDC and computer process claims. Again, there is a significant difference – chemistry is a non-deterministic technology in which the inventive application of chemical principles to real world problems must overcome an “infinity of permutations” which can only be overcome by extensive field testing.¹⁷⁹

Complexity

Compared to the industrial-era machines which the patent system was originally designed to protect, a software product is a highly complex construction. Whilst a physical machine may be made of thousands of individual components, a large computer program (such as an operating system) may contain millions of components.¹⁸⁰ The fact that each individual component may potentially be patented makes infringement possible on a massive scale. Such a software product as just described could, by definition, could potentially infringe millions of patents.

Another aspect of this high component-to-product ratio is that the patentability of such vast numbers of patentable components can lead to an administrative nightmare for national patent

¹⁷⁷ *In re Alappat* 33 F.3d 1526 (1994) at 1583 (emphasis added).

¹⁷⁸ *IBM v Commissioner of Patents* (1991) 33 FCR 218 (Chapter 3 at 121)

¹⁷⁹ Dukarich expressly notes chemical inventions as non-deterministic technologies. See Gary Dukarich, “Patentability of Dedicated information Processors and Infringement Protection of Inventions that Use Them” (1989) 29 *Jurimetrics Journal* 135 at 146-147.

¹⁸⁰ Richard Stallman, “The Danger of Software Patents,” Transcript of Speech Given at Cambridge University, March 2002, <<http://www.cl.cam.ac.uk/~mgk25/stallman-patents.html>> (13 September 2010) at [90].

Why is the patenting of software problematic?

offices. The USPTO put up stiff resistance to the introduction of software patents on the basis that allowing software patents would leave the patent office unable to handle the volume of patent applications which would be required. And that is exactly what has happened, with the USPTO acknowledging that the backlog of patents has grown remarkably:

[T]he volume and complexity of patent applications continues to outpace current capacity to examine them. The result is a pending – and growing – application backlog of historic proportions. Patent pendency ... now averages more than two years. In more complex art areas, such as data-processing technologies, average pendency now stands at more than 3 years.¹⁸¹

The obvious result of such a backlog is the increased likelihood that individual patent examinations will not be carried out thoroughly, resulting in an across-the-board drop in patent quality. When doubtful patents are passed, the costs of pursuing litigation to see them overturned (usually at the risk of infringement proceedings) means that often such patents are successfully enforced against competitors. Thus, the combined effect of the backlog and resultant grant of patents of dubious quality has given rise to regular criticisms in recent times that the patent system in the US is “broken”.¹⁸²

¹⁸¹ United States Patent and Trademark Office, *Performance and Accountability Report for Fiscal Year 2005*, <<http://www.uspto.gov/web/offices/com/annual/2005/2005annualreport.pdf>> (12 July 2010) at 4. In 1989, the number of total applications pending was 222,755, but by 2005 this number had reached 885,002: United States Patent and Trademark Office, “Fiscal Year 2009: Patent Applications Pending Prior to Allowance” <http://www.uspto.gov/web/offices/com/annual/2009/oai_05_wlt_03.html> (12 July 2010). By 2010, “[o]ver 1.2 million non-provisional patent applications [were] pending examination at the USPTO. Of those, more than 700,000 [had] not received even a preliminary examination.”: Dennis Crouch, “Unreasonable Applicant Delay and the USPTO Backlog” *Patently-O*, 9 July 2010, <<http://www.patentlyo.com/patent/2010/07/unreasonable-patent-applicant-delay-and-the-uspto-backlog.html>> (12 July 2010).

¹⁸² “Over the past several years, in newspaper articles and at hearings held by the Federal Trade Commission (FTC) and National Academy of Sciences (NAS), industry executives have complained in growing numbers that the patent system is broken.”: James Bessen and Michael J Meurer, “Chapter 1: The Argument in Brief” *Patent Failure: how judges, bureaucrats, and lawyers put innovators at risk* (2008) at 3. The FTC and NAS hearings, and the reports produced as a result are discussed in Carl Shapiro, “Patent System Reform: Economic Analysis and Critique” (2004) 19 *Berkeley Technology Law Journal* 1017; Kurt M Saunders, “Reforming the Patent System: Two Proposals” (2005) 4(1) *The Technology Report* <<http://www4.ncsu.edu/~baumerdl/The%20Technology%20Report/Revised.Reforming.the.Patent.System.pdf>> (13 July 2010). On the “broken” state of the patent system see also Adam B Jaffe & Josh Lerner, *Innovation and its discontents: how our broken patent system is endangering innovation and progress and what to do about it* (2004). The issue had also begun to attract the regular attention of the media. See for example Jeff Hecht, “Boy takes swing at US Patents” *NewScientist*, 17 April 2002, <<http://www.newscientist.com/article/dn2178-boy-takes-swing-at-us-patents.html>> (12 July 2010); Ryan Naraine, “When Software Patents Go Bad” *Internetnews.com*, 13 September 2002, <<http://www.internetnews.com/bus-news/article.php/1462741/When-Software-Patents-Go-Bad>> (12 July 2010); Lauren Weinstein, “Stop the Patent Process Madness” *Wired*, 21 October 2002, <<http://www.wired.com/politics/law/news/2002/10/55831>> (12 July 2010); Hal R. Varian, “Patent Protection Gone Awry” *New York Times*, 21 October 2004, <<http://www.nytimes.com/2004/10/21/business/21scene.html>> (12 July 2010); Sara Schaeffer, “Patent No. 6,004,596: Peanut Butter and Jelly Sandwich” *Wall Street Journal*, 5 April 2005 <<http://online.wsj.com/article/0,,SB111266108673297874,00.html>> (12 July 2010); Randall Stross, “Why Bill Gates Wants 3,000 New Patents” *New York Times*, 31 July 2005, <<http://nyti.ms/hzgnGC>> (12 July 2010); Tuan Nguyen, “Red Hat Says U.S. Patent System Is Broken” *Tom’s Guide*, 8 April 2008, <<http://www.tomsguide.com/us/redhat-patent-lawsuit,news-961.html>> (12 July 2010); L Gordon Crovitz, “Patent Gridlock Suppresses Innovation” *Wall Street Journal*, 14 July 2008, <<http://online.wsj.com/article/SB121599469382949593.html>> (12 July 2010).

Chapter 4: Why Software Patents are a Problem

Reuse

It will be recalled from Chapter 1 that software has developed in an incremental fashion, from machine code to assemblers, assemblers to compilers and interpreters, and so on. This reliance on reuse has created an industry built around that fact, which affords further reasons why a categorical exclusion of software from the patent regime needs to be considered. Although “most inventions represent improvements on some existing article, process or machine”,¹⁸³ because reuse lies at the very heart of software development, the problems created are in a sense unique. The main problems which improperly awarded patents create for the software industry are outlined below.

Software innovations are largely cumulative

The low barriers to entry in software development mean that the market is massively decentralised. With the cheapest computer, and an Internet connection, it is possible for anyone to download a compiler or interpreter and start learning how to program. This decentralisation leads to independent repetition, “because programmers use similar, if not identical, software and hardware tools to tackle common needs”.¹⁸⁴ Part of the reason for the similarity in tools is the influence of lower layers of the abstraction stack setting a common framework for development at the upper levels. Because of this repetitiveness, and perhaps due in part to the sharing ethos developed in the early era computer science, and the importance of familiarity to users, “[p]rogrammers commonly adopt software design elements - ideas about how to do particular things in software - by looking around for examples or remembering what worked in other programs”.¹⁸⁵ As such, most new developments of the technology tend to be cumulative, grain-sized innovations rather than big leaps forward.

The cumulative nature of software development has been consistently under-emphasised by the courts and patent offices, meaning that large rewards (in the form of patent licences) are available as the result of minimal research. The lure of large profits thus encourages opportunists to obtain and enforce broad patents at the expense of innovation in the field.¹⁸⁶

The software industry relies on openness

Since its origins in the computer science departments of universities, openness and collaboration has had an influential effect on the state of the art in software. This was probably a result of the tediousness of development in machine code and assembly making it clear that progress depended on reuse. At the earliest stages, computer scientists shared the source code of software innovations, either directly or through publication in freely available journals. This ethos

¹⁸³ *General Electric Company v Wabash Appliance Corporation* 304 U.S. 364, 368 (1938)

¹⁸⁴ Ben Klemens, “New legal code: copyrights should replace software patents” (2005) *IEEE Spectrum Magazine*, <<http://spectrum.ieee.org/computing/software/new-legal-code>> (18 September 2010).

¹⁸⁵ See Pamela Samuelson et al, “A Manifesto Concerning the Legal Protection of Computer Programs,” (1994) 94 *Columbia Law Review* 2308 at 2330-2331.

¹⁸⁶ See William W Fisher III, “Intellectual Property and Innovation: Theoretical, Empirical, and Historical Perspectives” (2001) 37 *Industrial Property, Innovation, and the Knowledge-based Economy*, *Beleidsstudies Technologie Economie*, at 18.

Why is the patenting of software problematic?

of openness amongst software developers continued even as commercial players began to get involved. For example, the early history of the Unix operating system for a long time involved an open collaboration between AT&T and the University of California at Berkeley.¹⁸⁷

Perhaps the most important form of openness in modern software development, however, is the Free and Open Source Software (FOSS) model, which provides a mechanism for global, collaborative innovation based on principles of freedom and sharing.¹⁸⁸ FOSS represents an important subset of the software industry, because it embraces an alternative path to innovation and global collaboration which furthers the public interest in the development and disclosure of new technologies – the same goal claimed by the patent system.¹⁸⁹

The continued importance of openness to innovation in software development is at odds with the award of monopoly rights. That software patents increase innovation in the field is also challenged by impressionistic evidence of the US software industry, which suggests that innovation levels were already high before software patenting began, and have not increased since.¹⁹⁰

FOSS developers face the same problems as all software developers, as outlined above, but there are a number of ways in which FOSS projects are more vulnerable than their proprietary counterparts. Firstly, FOSS development is in most cases undertaken by small service-oriented companies who make their money out of implementation and administration of the software on customer projects, or in many cases volunteers.¹⁹¹ Thus the problems of SMEs discussed above are particularly relevant to the ongoing viability of FOSS projects. The problem is multiplied however as FOSS projects generally lack a central body to fund and harvest a large patent portfolio, or to negotiate licensing agreements to cover project participants en masse.

Further, FOSS projects harness the collaborative power of the Internet which allows co-operation on projects on a global scale exceeding even that of the software industry monoliths.¹⁹² The ge-

¹⁸⁷ For a more detailed history of the BSD project, see Marshall K McKusick, "Twenty Years of Berkeley Unix: From AT&T-Owned to Freely Redistributable" in Chris Di Bona and Sam Ockman (eds) *Open Sources: Voices from the Open Source Revolution* (O'Reilly & Associates, 1999).

¹⁸⁸ For more on the nature of FOSS, see Kenneth Wong and Phet Sayo, "Free/Open Source Software: A General Introduction," *International Open Source Network*, 2004 <<http://www.iosn.net/foss/foss-general-primer/>> (18 September 2010).

¹⁸⁹ "Accelerating the moment at which knowledge is widely available is consistent with patent policy's design to bring inventions into the public domain for the public benefit.": Robin Feldman, "The Open Source Biotechnology Movement: Is It Patent Misuse?" (2004) 6(1) *Minnesota Journal of Law, Science & Technology* 117 at 120. Feldman also notes that open source systems encourages downstream innovation and decreases the likelihood of patent thickets and avoids the short term restriction of supply which is a feature of traditional patent licensing. However, Feldman notes that the open source licence may reduce the economic reward available to downstream inventors, which is inconsistent with patent policy.

¹⁹⁰ William W Fisher III, "Intellectual Property and Innovation: Theoretical, Empirical, and Historical Perspectives" (2001) 37 *Industrial Property, Innovation, and the Knowledge-based Economy*, *Beleidsstudies Technologie Economie* at 24-25.

¹⁹¹ The author bases this proposition on his personal experience with the Zope and Plone project communities (see <http://zope.org/> and <http://plone.org/> respectively). Empirical evidence gathered in relation to Embedded Linux also supports this view, with 139 of 259 survey participants working for organisations with less than 50 people, and 81 participants working for companies with 200 or more staff. See Joachim Henkel and Mark Tins, "Munich/MIT Survey: Development of Embedded Linux" (Institute for Innovation Research, Technology Management and Entrepreneurship, University of Munich, 10 May 2004) <http://www.linuxfordevices.com/files/misc/MunichMIT-Survey_Embedded_Linux.pdf> (5 September 2011) at 7.

¹⁹² Russ Roberts, "Why Linux is Wealthier than Microsoft" *BusinessWeek Online*, 19 November 2003 <http://yahoo.businessweek.com/technology/content/nov2003/tc20031119_9737.htm> (18 September 2010).

Chapter 4: Why Software Patents are a Problem

ographical distribution of project participants together with the ability of any party to “fork” the code,¹⁹³ a central tenet of the open source paradigm, illustrates the distributed nature of FOSS development in which no central body has ultimate control. Sometimes a project will have one or more leaders,¹⁹⁴ or a non-profit organisation which holds the intellectual property and hosts development services,¹⁹⁵ but generally speaking, participants in such projects are widely distributed around the globe, and the levels of participation, and responsibility for various aspects changes from time to time. This may be the key to the success of these projects, but the downside is that legal responsibility for patent infringement is likely to be directed at individual contributors and even users of the software. Whilst some may suggest that the sheer difficulty of pursuing a distributed target makes enforcement less likely,¹⁹⁶ many open source advocates have long felt that patents represent the biggest threat to this form of software development.¹⁹⁷

The threat of even high profile FOSS projects being shut down by the spectre of patent litigation is a real one. In a recent study undertaken by Open Source Risk Management, it was found that the Linux kernel potentially infringes 283 US software patents which have yet to be tested by the courts.¹⁹⁸ Stallman has estimated that the Linux kernel represents only 0.25% of a complete GNU/Linux system,¹⁹⁹ suggesting a possible total of between 30,000 and 300,000 possible infringements overall.²⁰⁰ Although a “a third of the patents are owned by Linux backers, including Hewlett-Packard, IBM, Novell, and Oracle, which are unlikely to assert claims,” such a precarious position is a cause for concern as it only takes one patent to shut down a project.²⁰¹ The controversial decisions by some GNU/Linux distributors, including Novell, Xandros and Linspire to enter into a “Patent Cooperation Agreement” with Microsoft illustrates that many feel the threat is very real.²⁰²

¹⁹³ Code forking is a consequence of the liberal modification and redistribution rights given in OSS licences. If any one party is unhappy with the direction a project is taking, they can take a copy of the source code and use it to launch their own project. See Wikipedia, “Code Forking” <http://en.wikipedia.org/wiki/Code_forking> (18 September 2010).

¹⁹⁴ For example, Linus Torvalds is the project leader for the Linux kernel project, and Guido van Rossum is the leader of the Python language project.

¹⁹⁵ For example the Apache Software Foundation (see <<http://apache.org/>>) or the Mozilla Foundation (see <<http://mozilla.org/>>) host the project website, version control and mailing lists which allow project collaboration in addition to holding IP rights in the project code.

¹⁹⁶ Peter Williams, “Patent threat to open source is limited” *vnunet.com*, 8 September 2004 <<http://www.pcw.co.uk/vnunet/news/2125807/patent-threat-open-source-limited>> (18 September 2010). On the other hand, the Recording Industry Association of America are certainly having a good shot at pursuing similarly distributed file-sharers with copyright infringement suits. See Robert Jaques, “RIAA Launches P2P file sharing legal blitz” *vnunet.com*, 19 November 2004 <<http://www.v3.co.uk/vnunet/news/2126208/riaa-launches-p2p-file-sharing-legal-blitz>> (18 September 2010).

¹⁹⁷ See for example Charles Babcock and Larry Greenemeier, “Open Source Stress” *Information Week*, 9 August 2004 <<http://www.informationweek.com/story/showArticle.jhtml?articleID=26806464>> (18 September 2010).

¹⁹⁸ Dan Ravicher, “Position Paper: Mitigating Linux Patent Risk” *Open Source Risk Management*, 2 August 2004 <http://www.osriskmanagement.com/pdf_articles/linuxpatentpaper.pdf> (4 September 2011) at 1.

¹⁹⁹ This is what people mistakenly refer to as the Linux operating system, which Stallman insists is more correctly called GNU/Linux. See for example, Richard Stallman, “Patent absurdity” *The Guardian*, 20 June 2005 <<http://www.guardian.co.uk/technology/2005/jun/20/comment.comment>> (4 September 2005).

²⁰⁰ Richard Stallman, “Patent absurdity” *The Guardian*, 20 June 2005 <<http://www.guardian.co.uk/technology/2005/jun/20/comment.comment>> (4 September 2005).

²⁰¹ For example, the preamble to the GNU General Public Licence has for a decade recognised that “any free program is threatened constantly by software patents”. See Free Software Foundation, “GNU General Public License” 16 February 1998 <<http://www.gnu.org/copyleft/gpl.html>> (18 September 2010).

²⁰² See for example Microsoft Legal and Corporate Affairs, “Patent Cooperation Agreement - Microsoft and Novell Interoperability Collaboration” *Microsoft.com*, 2 November 2006

How does the award of patents affect the software industry?

Summary

It has been seen that the related concepts of abstraction, or abstractness, and context, or characterisation, form the primary theoretical dimensions of the software patent problem. The abstract nature of the subject matter, potentially holds the key to understanding why software should not be patentable. But attempts to address this abstractness, through appeals to traditional exclusions such as mental steps, intellectual information and abstract ideas, have faltered in the absence of a strong theoretical foundation to underpin them. At the other end of the spectrum, the physicality of an claimed invention (and as a corollary, its absence in abstract subject matter) has a strong historical and theoretical foundation. However, the case law demonstrates that software has been regularly mischaracterised, and its context ignored, so as to avoid difficult ontological questions as to the nature of a particular set of software-related claims.

The remaining aspects of the nature of software, complexity and reuse, give rise to their own practical problems. The complexity of software gives rise to an over-supply of patentable components which threaten both to overload patent offices, and increase the probability that an independently developed product infringes on multiple patents. The centrality of reuse to the software development process similarly operates to reduce the grain size of software innovations, and increase the likelihood of infringement. Reliance on reuse also means that the software industry's culture of openness, and alternative innovation paradigms such as FOSS stand to be greatly disrupted by the award of monopoly rights, lessening the public interest in new innovation.

Any proposed solution of the software patent problem will therefore need to address these issues.

3 How does the award of patents affect the software industry?

In addition to the problems which the nature of software causes in patent law, there are a number of other considerations which are relevant to the issue of whether software ought to be patentable subject matter. These are discussed below.

Poor documentation means more trivial patents

Programmers are notoriously lousy at documenting the programs they write. This is perhaps because of the source code *is* the documentation, or perhaps because compared to the active problem solving involved in writing code, writing documentation is just plain boring.²⁰³ Even

<http://www.microsoft.com/about/legal/en/us/IntellectualProperty/IPLicensing/customer covenant/msnovellcollab/patent_agreement> (4 September 2011); Rich Lehrbaum, "Linspire, Microsoft in Linux-related deal" *DesktopLinux.com*, 13 June 2007 <<http://desktoplinux.com/news/NS9642338710.html>> (4 September 2011); Xandros Inc, "Microsoft, Xandros Broad Collaboration Agreement Extends Bridge Between Commercial Open Source and Microsoft Software" on *xandros.com*, 4 June 2007 <http://www.xandros.com/news/press_releases/xandros_microsoft_collaborate.html> (4 September 2011).

²⁰³ For a representative range of views on the issue, see "Why Do Programmers Hate Documenting?" <<http://discuss.fogcreek.com/joelonsoftware1/default.asp?cmd=show&ixPost=35336&ixReplies=61>>

Chapter 4: Why Software Patents are a Problem

when documentation is properly written however, the ubiquity of computers, the global nature of the industry, and the absence of any centralised knowledge repository (such as industry journals) means that it is very difficult to adequately represent the entire state of the art in documentary form at any point in time.²⁰⁴ The absence of documentation presents serious problems to the patent examination process, as proper documentation is critical to the application of patentability standards. Without a complete prior art base, it is impossible for patent examiners, who are rarely software developers, to adjudge whether an alleged invention is truly novel, and whether such an advance would have been obvious to a professional working in the field (the non-obvious requirement). The result of poor documentation is thus that trivial advances over the state of the art will regularly be elevated to the status of patentable invention.²⁰⁵

The software market moves too fast for the patent regime

A patent grant involves the award of monopoly rights to individual entities, typically for a 20 year period. This is an eternity in the fast-paced software industry, where a typical product life-cycle is around 3-5 years.²⁰⁶ Because patents are construed as broadly as the language of their claims will allow, however, patent holders may be able to assume control over technological developments which were far beyond the state of the art at the time the patent was granted.

Even the application process is too slow. As noted above, the waiting period for the issue of a software patent is over 3 years.²⁰⁷ The application will remain secret for some, if not all, of that period.²⁰⁸ Given the independent repetition which characterises the industry, it thus

²⁰⁴ As noted in the Introduction to this thesis, in 2008 there were around 1.3 million computer systems software engineers and programmers in the US. That number was projected to increase by 21%, to around 1.6 million, by 2018: United States Bureau of Labor Statistics, "Computer Software Engineers and Computer Programmers" in *Occupational Outlook Handbook, 2010-11 Edition* <<http://www.bls.gov/oco/ocos303.htm>> (3 September 2011). Based on 2002 or later data on the relative size of the software industries in Brazil, China, India, Ireland, Israel, Japan and Germany, it is estimated that the number of programmers in those countries would be in the region of 660,000. See Asish Arora and Alfonso Gambardella, "The Globalization of the Software Industry: Perspectives and Opportunities For Developed and Developing Countries" (National Bureau of Economic Research Working Paper 10538, June 2004) <<http://www.nber.org/papers/w10538>> (18 September 2010). The actual number of people writing software is likely to be much higher, as software may also be written by computer software engineers; computer scientists and database administrators; and computer systems analysts who are all classified separately by the Bureau. This estimate also fails to capture volunteer programmers, students and those working in seemingly unrelated disciplines, for example physicists who write software to manage and conduct their research.

²⁰⁵ This is the criticism typically leveled at the Amazon 1-click patent, which many developers consider to be an obvious application of cookie technology. For an enlightening discussion of the controversy, see Tim O'Reilly, "Ask Tim: Open Source, Patents & O'Reilly" *O'Reilly Network*, 28 February 2000 <http://oreilly.com/pub/a/oreilly/ask_tim/2000/amazon_patent.html> (18 September 2010)

²⁰⁶ For an overview see Wikipedia, "Moore's Law" <http://en.wikipedia.org/wiki/Moore%27s_law> (18 September 2010). For a more in-depth analysis, see Ilkka Tuomi, "The Lives and Death of Moore's Law" (2002) 7(11) *First Monday* <<http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/1000/921>> (18 September 2010).

²⁰⁷ See the discussion of the backlog at the USPTO on page 172 above. Note that for Tech Center 2100 (Computer Architecture, Software & Information Security), the pendency for *first action* on a patent application at the USPTO is 29 months, with total average pendency standing at 40 months. See United States Patent Office, "Patent Pendency Statistics - FY09" <<http://www.uspto.gov/patents/stats/patentpendency.jsp>> (18 September 2010).

²⁰⁸ An ordinary application to a national patent office will not be published until the patent is granted. However, if it is an international application under the terms of the *Patent Co-operation Treaty*, [1980] ATS 6 (entered into force on 24 January 1978), Article 21(b) requires that, subject to certain exceptions, the international publication is to be "effected promptly" after 18 months.

How does the award of patents affect the software industry?

becomes likely that an entirely independent yet infringing software product could be designed, implemented, brought to market, and even replaced by a second generation version before any knowledge of a relevant patent was available. Infringement however, runs from the date of application, meaning substantial royalties would be owed. In this scenario, the product's developer reaped none of the benefit of the patent and bears the entire burden of establishing a successful market for their product, yet the patent holder is able to hold the product developer's business hostage until royalties and/or a suitable licence are negotiated. Such royalties are thus an unwarranted tax on independent innovation.

Network effects amplify the power of patent holders

Network effects are common in software circles, where having a trained base of users for a particular product creates barriers to entry for competitor products. A network effect refers to "the social advantages that arise when all of the users of a particular type of technology adhere to the same standards and thus can share their work and move easily between machines and businesses".²⁰⁹ A simple example is the Microsoft Word file, which is a de facto standard for storing and exchanging word processor documents.

Network effects may also arise as a result of the adoption by the industry of a particular standard. For example, the TCP/IP protocol is the standard for the transmission of digital information over computer networks, particularly the Internet. Standards are generally used to encourage interoperability and collaboration in a technologically optimal way.

Network effects amplify the ability of their owners to control a market, by removing the bargaining power of competitors. For example, incorporating a patented technology into a standard allows the patent holder to enforce a licence fee from not only all competitors needing their products to interoperate with the standard, but also users of competing products.²¹⁰ This is because the patent cannot be "invented around" without deviating from the standard.²¹¹

Programmers hate software patents

If patent law's social contract is working as it should, then one would expect that those most likely to benefit from the disclosure of new technologies, namely programmers, would value patents as an important source of information. In fact most programmers believe that software

²⁰⁹ William W Fisher III, "Intellectual Property and Innovation: Theoretical, Empirical, and Historical Perspectives" (2001) 37 *Industrial Property, Innovation, and the Knowledge-based Economy*, Beleidstudies Technologie Economie at 25.

²¹⁰ Technically, use of the product would fall within the scope of the patent holder's right to exploit the invention. It is assumed however that a patent holder is unlikely to sue its own customers.

²¹¹ It is to get around network effects that patent pools are often created. Depending on the market power of those who participate in the pool, such pools may have either pro-competitive or anti-competitive effects. See Steven C Carlson, "Patent Pools and the Antitrust Dilemma" 16 *Yale Journal on Regulation* 359 (1999); Jeanne Clark et al "Patent Pools: A Solution to the Problem of Access in Biotechnology Patents" (2001) 20(4) *Biotechnology Law Report* 607; Robin Feldman, "The Open Source Biotechnology Movement: Is It Patent Misuse?" (2004) 6(1) *Minnesota Journal of Law, Science & Technology* 117 at 166-167.

Chapter 4: Why Software Patents are a Problem

patents “significantly hamper their work”.²¹² The League for Programming Freedom for example has been arguing for their abolition since at least 1991.²¹³ Unsurprisingly, the voices of key of free and open source software figures such as Richard Stallman,²¹⁴ Linus Torvalds,²¹⁵ and Bruce Perens²¹⁶ are unanimous in arguing that the only solution to the problems which software patents cause software developers is to abolish software patents altogether. There is also empirical²¹⁷ and anecdotal evidence²¹⁸ to suggest that proprietary software developers would happily do without software patents. Why is this?

Firstly, despite the prevalence in the public consciousness of large US firms such as IBM, Microsoft and Oracle, the global software industry is largely populated by small-to-medium enterprises (SMEs).²¹⁹ In Australia for example, a 2005 study found that 99.6% of Australian computer software and services companies contained less than 100 employees.²²⁰ These small players are particularly vulnerable to the problems software patenting creates. Their limited budgets means that they can not afford the license fees, the costs involved in challenging invalid patents or defending against infringement. It is they who feel the pain of software patenting most sorely.²²¹ It is little wonder then that SMEs largely threw their weight behind the anti-software patenting camp in the recent battle to prevent the legitimisation of software patenting in the EU.²²²

The software giants are generally able to limit the likelihood of infringement proceedings by maintaining large patent portfolios. These portfolios are reminiscent of the “mutually assured destruction”²²³ policy of the superpowers in the Cold War, in that any would-be litigant is

²¹² William W Fisher III, “Intellectual Property and Innovation: Theoretical, Empirical, and Historical Perspectives” (2001) 37 *Industrial Property, Innovation, and the Knowledge-based Economy*, Beileidsstudies Technologie Economie at 25.

²¹³ See League for Programming Freedom, “Against Software Patents” 28 February 1991 <<http://www.progfree.org/Patents/against-software-patents.html>> (5 September 2011).

²¹⁴ Richard Stallman, “The Danger of Software Patents” 2004 Cyberspace Law and Policy Seminar (audio recording), Sydney, 14 October 2004 <http://www.bakercyberlawcentre.org/2004/talks/Stallman_1_small.ogg> (18 September 2010) (Part 1) and <http://www.bakercyberlawcentre.org/2004/talks/Stallman_2_small.ogg> (18 September 2010) (Part 2).

²¹⁵ See Linus Torvalds and Alan Cox, “Open Letter on Software Patents from Linux Developers” 21 September, 2003 <http://www.effi.org/patentit/patents_torvalds_cox.html> (5 September 2011); Robert McMillan, “Torvalds joins in anti-patent attack” *Techworld*, 2 February 2005, <<http://news.techworld.com/applications/3059/torvalds-joins-in-anti-patent-attack/>> (18 September 2010).

²¹⁶ Bruce Perens, “Public Policy Area: Software Patents” <<http://perens.com/policy/software-patents/>> (18 September 2010); Bruce Perens, “The Monster Arrives: Software Patent Lawsuits Against Open Source Developers” <<http://technocrat.net/d/2006/6/30/5032/>> (18 September 2010).

²¹⁷ See Effy Oz, “Acceptable Protection of Software Intellectual Property: A Survey of Software Developers and Lawyers.” (1998) 34(3) *Information & Management* 161.

²¹⁸ The selection of anti-software-patent quotes collated on this page read like a virtual who’s-who of the US Software Industry. See Foundation for a Free Information Infrastructure, “Quotations on Software Patents,” <<http://eupat.ffii.org/vreji/quotes/index.en.html>> (18 September 2010).

²¹⁹ See Pamela Samuelson et al, “A Manifesto Concerning the Legal Protection of Computer Programs,” (1994) 94 *Columbia Law Review* 2308 at 2377.

²²⁰ Centre for Innovative Industry Economic Research Consortium, “Executive Summary”, *The Australian software industry and vertical applications markets: Globally competitive and domestically undervalued*, <http://www.dbcde.gov.au/_data/assets/pdf_file/0020/36461/Executive_summary.pdf> (5 September 2011).

²²¹ On this point see Puay Tang, John Adams and Daniel Paré, “Patent protection of computer programmes” (Final Report submitted to the European Commission, 2001).

²²² Foundation for a Free Information Infrastructure, “Alliance of 2,000,000 SMEs against Software Patents and EU Directive,” 16 September 2003, <<http://eupat.ffii.org/papers/eubsa-swpat0202/ceapme0309/>> (18 September 2010).

²²³ See Wikipedia, “Mutually Assured Destruction,” <http://en.wikipedia.org/wiki/Mutually_assured_destruction> (18 September 2010).

Is this a subject matter issue?

discouraged from initiating proceedings lest they should be hit with a counter-claim for infringement themselves. However, software patenting nevertheless impacts negatively on the larger players. Maintaining such a strategy consumes time and money. Robert Barr, vice president and head of the patent department of Cisco Inc, in a frank admission to the FTC, put it this way:

The time and money we spend on patent filings, prosecution, and maintenance, litigation and licensing could be better spent on product development and research leading to more innovation. But we are filing hundreds of patents each year for reasons unrelated to promoting or protecting innovation.²²⁴

Defensive portfolios also do little to discourage so-called patent trolls – companies who do not engage in software development, and whose sole source of income depends on the opportunistic enforcement of the patents they acquire.²²⁵

4 Is this a subject matter issue?

One point which emerges from the survey of EPC jurisprudence is that the objectionability of software patents is considered by the EPO Board of Appeal to be an inventive step issue, not a subject matter issue. For example, the Enlarged Board in G3/08 noted that:

While the Enlarged Board is aware that this rejection for lack of an inventive step rather than exclusion under Article 52(2) EPC is in some way distasteful to many people, it is the approach which has been consistently developed since [the *IBM Twins*] and since no divergences from that development have been identified in the referral we consider it not to be the function of the Enlarged Board in this Opinion to overturn it.²²⁶

A similar attitude is evident in the US Federal Circuit decision in *Diehr*, who in rejecting the *Flook* “point of novelty” approach to determining patentable subject matter stated that

[the question] of whether a particular invention is novel is wholly apart from whether the invention falls into a category of patentable subject matter.²²⁷

The majority in *Bilski v Kappos* also seem to put much faith in the other requirements for patentability:

²²⁴ Robert Barr, Statement to the Joint Federal Trade Commission and Department of Justice Hearings on Economic Perspectives on Intellectual Property, Competition and Innovation, 28 February 2002, <<http://www.ftc.gov/opp/intellect/barrrobert.doc>> (5 September 2011).

²²⁵ For more on how patent trolls operate, see Wikipedia, “Patent Troll,” <http://en.wikipedia.org/wiki/Patent_troll> (18 September 2010).

²²⁶ G3/08 *Programs for computers* (Opinion of 12 May 2010) at [10.13].

²²⁷ *Diamond v Diehr* 450 US 175 (1981) at 190. See also *In re Bernhart* 417 F.2d 1395 (1969) (Chapter 3 at 3)

Chapter 4: Why Software Patents are a Problem

The §101 eligibility inquiry is *only a threshold test*. Even if a claimed invention qualifies in one of the four categories, it must also satisfy “the conditions and requirements of this title,” §101(a), including novelty, see §102, nonobviousness, see §103, and a full and particular description, see §112.²²⁸

Given the difficulty of the area, it may not be surprising that courts have sought to avoid subject matter issues and rely instead on the more instant issues of whether claimed inventions are new. On the other hand, the UK Court of Appeal *Aerotel* illustrate the problems with moving the issue to the inventive inquiry step by way of example:

Consider for instance the following:

- i) a claim to a book, e.g. to a book containing a new story the key elements of which are set out in the claim;
- ii) a claim to a standard CD player or iPod loaded with a new piece of music.

Everyone would agree that the claims must be bad – yet in each case as a whole they are novel, non-obvious and enabling. To deem the new music or story part of the prior art (the device of *Pension Benefits* and *Hitachi*) is simply not intellectually honest. And, so far as we see, the *Microsoft* approach, which discards that device, would actually lead to patentability.²²⁹

Christie and Syme²³⁰ identify a further issue which may render inventiveness an unsuitable lens for considering this issue. The test for inventiveness takes place in the context of the relevant art.²³¹ When a computer program forms the substance of the claims, then the relevant art is likely to be computer programming. However, the nature of computer programming is such that it often involves the automation of processes in other areas. To use Christie and Symes’ example,²³² when a mathematical algorithm is translated to computer-automated form,

²²⁸ *Bilski v Kappos* 561 US ____ (2010) (Kennedy J slip opinion) at 2 (emphasis added). Note that the USPTO’s post-Bilski interim guidelines have picked up on this remark, and remind examiners “that §101 is not the sole tool for determining patentability ... Therefore, examiners should avoid focusing on issues of patent-eligibility under § 101 to the detriment of considering an application for compliance with the requirements of §§ 102, 103, and 112, and should avoid treating an application solely on the basis of patent-eligibility under § 101 except in the most extreme cases”: United States Patent and Trademark Office “Interim Guidance for Determining Subject Matter Eligibility for Process Claims in View of *Bilski v Kappos*” 75 Fed Reg 43922 (27 July 2010) at 43923-43924. Cf *Bilski v Kappos* per Stevens J – see n235 below.

²²⁹ *Aerotel Ltd. v. Telco Holdings Ltd. and in the Matter of Macrossan’s Application* [2006] EWCA Civ 1371; [2007] RPC 7 at [27].

²³⁰ Andrew Christie & Serena Syme, “Patents for Algorithms in Australia,” (1998) 20 *Sydney Law Review* 517.

²³¹ Christie and Syme are discussing the Australian approach to inventive step, but their comments are equally applicable to the US and UK approaches. As to the UK, see the case of *See Pozzoli Spa v BDMO SA & Anor* [2007] EWCA Civ 588 (22 June 2007) where satisfaction of the requirement is determined by reference to the notional “person skilled in the art”. This conception is also at the heart of the EPO’s problem-solution approach. See T154/04 *Duns/Method of estimating product distribution* [2008] OJ EPO 46 at Reasons 5(G): “for the purpose of the problem-and-solution approach, the problem must be a technical problem which the skilled person in the particular technical field might be asked to solve at the relevant priority date.” Similarly the US approach is tested by reference to the “personal having ordinary skill in the art” §103 *Patent Act 1952* (US). For the most recent interpretation given to this standard see *KSR International Co. v. Teleflex Inc., et al.* 550 U.S. 398 (2007).

²³² See Andrew Christie & Serena Syme, “Patents for Algorithms in Australia,” (1998) 20 *Sydney Law Review* 517 at 553.

How should the analysis proceed?

it will not be relevant that the algorithm itself has been in use in the field of mathematics for hundreds of years if it has not been used in computer programming before.

Further, the determination of the issue is further hamstrung by the incomplete prior art due to the lack of documentation of software inventions, as noted above. Whether the prior art can be sufficiently determined is not an issue which falls to be considered within the confines of the inventive step test, but would, it is submitted, be an appropriate consideration in determining the inherent patentability of a category of inventions likely to suffer from this malady. This is because inherent patentability is the “first and most fundamental”²³³ threshold which any patent application must surmount is the patentable subject matter inquiry. Whereas the other requirements focus on the technical aspects of the invention, the patentable subject matter inquiry is intended to establish the *prima facie* availability of patents for subject matters suitable for the award of patents.²³⁴ As such, this inquiry can (and it is submitted should) address a broader range of considerations.

In other words, “the requirements of novelty, nonobviousness, and particular description [can not] pick up the slack.”²³⁵ That notions of newness and inventiveness may be touched on by the inherent patentability inquiry is not a new development – the history of excluding analogous uses from patentability at the very least confirms that.²³⁶ But beyond this, the nature of inherent patentability as a threshold issue, subject as it is to the need to determine what the applicant has invented,²³⁷ means that some consideration of what is said to be the invention, in a broad sense, will ordinarily be required. To hold that the characterisation process unnecessarily overlaps with other requirements is to ignore both the history and the purpose of the inquiry.

5 How should the analysis proceed?

From the parameters of the software patent problem just discussed, a number of issues have been identified. Firstly, the lack of a satisfying account as to why abstract ideas are non-patentable is of concern. Secondly, it has been shown how the lack of a theoretical underpinning for exclusions such as the mental steps doctrine have resulted in their erosion over time. Taken

²³³ *Patentable Subject Matter: Issues Paper* (2008) <<http://www.acip.gov.au/library/Patentable%20Subject%20Matter%20Issues%20Paper.pdf>> (accessed 13 November 2008) at 1.

²³⁴ Justine Pila, “Response to the Australian Government Advisory Council on Intellectual Property’s Request for Written Comments on Patentable Subject Matter” *Oxford Legal Studies Research Paper No. 37/2008*, September 19, 2008 <http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1274102> at (ref?).

²³⁵ *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3238, n 5 per Stevens J. His Honour also noted the “substantial academic debate, . . . about whether the normal process of screening patents for novelty and obviousness can function effectively for business methods. The argument goes that because business methods are both vague and not confined to any one industry, there is not a well-confined body of prior art to consult, and therefore many “bad” patents are likely to issue, a problem that would need to be sorted out in later litigation.”: at 3256 n55 (citing in support Rochelle Dreyfuss, “Are Business Methods Patents Bad for Business?” (2000) 16 *Santa Clara Computer & High Technology Law Journal* 263 at 268-270; Rebecca Eisenberg, “Analyze This: A Law and Economics Agenda for the Patent System” (2000) 53 *Vanderbilt Law Review* 2081 at 2090; Robert Merges “Property Rights for Business Concepts and Patent System Reform” (1999) 14 *Berkeley Technology Law Journal* 577 at 589-590). This accords with the criticisms just levelled at software patents.

²³⁶ See the discussion in Chapter 2 on page 47, noting *Losh v Hague* (1838) 1 Web Pat 202 (NP); *Harwood v Great Northern Railway* (1865) 11 ER 1488; *Muntz v Foster* (1843) 2 WPC 93 (Chapter 2 at 47). See also *Crane v Price* (1842) 134 ER 239 (Chapter 2 at 47).

²³⁷ See n116 above.

Chapter 4: Why Software Patents are a Problem

together, these two concerns suggest that the failure to properly articulate the reasons for excluding abstract ideas may lead to this exclusion itself being eroded. Given that, in the US at least, this exclusion may be all “that is left to stand between all conceivable human activity and patent monopolies”,²³⁸ such a result would be disastrous.

It is argued that many of the problems just reviewed can be explained by the limitations of a narrow, economic/utilitarian approach to patent law theory, and thus to the patentable subject matter inquiry. Many of the issues reviewed do not fit neatly within such an analysis. Therefore if the software patent problem is to be solved, a new approach is required.

Given the isomorphism between mathematics and software outlined in Chapter 1, it might be expected that the non-patentability of the former would be dispositive of the status of the latter. Yet it seems that the relationship between software and mathematics is poorly understood. The dominant approach is that such as in *Benson*, which looking at the problem by reference to the ‘algorithm’, which tries to suggest that some software is mathematical, whereas some is not. As discussed in Chapter 3, such a distinction is illusory.²³⁹

Although it is generally understood that mathematics is not patentable, the reasons for this conclusion have never been identified. The closeness of software and mathematics means that such an understanding is more than an issue of passing interest. Mathematics is a discipline as old as human-kind. Yet in the 400 year history of patent law, it has received only tangential consideration. Given the closeness of the relationship between software and mathematics it submitted that the nature of mathematics, and its patentability are questions of central importance to the patentability of software related inventions.

Having reviewed patent law from the earliest systems to the current position, and having seen the nature of the problems created by the current approach to determining the patentability of computer-related inventions, it is [contended that there is value in taking a different approach to determining the patentability of software](#).²⁴⁰ Thus it is proposed to pursue the patentability of software by reference to the patentability of mathematics.

6 Conclusion

This chapter has explored the parameters of the software patent problem, informed by the historical foundations of patentable subject matter jurisprudence, and modern interpretations of such jurisprudence to this unique and confounding subject matter. The way in which courts and tribunals in every jurisdiction have been confounded in their attempts to compellingly explain why it is that software should or should not be patentable leads to the conclusion that a new approach is required. It has been posited that the reason why software’s patentability is

²³⁸ *Bilski v Kappos* 561 US ____ (2010) (Stevens J slip opinion) at 13, note 5.

²³⁹ See Chapter 3 at 17, in particular the quote extract from Allen Newell, “Response: The Models Are Broken, The Models Are Broken,” (1986) 47 *University of Pittsburgh Law Review* 1023, wherein Newell notes that “there is an underlying identity between the numerical and the nonnumerical realms that will confound any attempt to create a useful distinction between them”: at 1024-1025.

²⁴⁰ [It will be recalled that a new approach was foreshadowed in the Introduction, where the benefits of, and proposed limitations upon this new approach were discussed. See Introduction, at 5.](#)

Conclusion

a vexed question is because of its misunderstood relationship with mathematics, and that the nature of mathematics, and the reasons for its non-patentability may lead to a solution.

The journey towards that solution must begin with a detailed consideration of the nature of mathematics. As will be seen in the next chapter, the avoidance of mathematics by patent lawyers may be an informed choice. There is much dispute, even among mathematicians, as to what mathematics is. It is not proposed to solve that question, but it is expected that by reviewing the disparity of theories as to the nature of mathematics, a theory of its non-patentability might be developed.

The Nature of Mathematics

“Courts have used the terms ‘mathematical algorithm,’ ‘mathematical formula,’ and ‘mathematical equation,’ to describe types of nonstatutory mathematical subject matter without explaining whether the terms are interchangeable or different. Even assuming the words connote the same concept, there is considerable question as to exactly what the concept encompasses.”¹

It has been shown that the current approach to patentable subject matter is problematic from both practical and theoretical perspectives. Whilst the ultimate aim of this thesis is to argue that software patent is not patentable, this chapter begins that journey by looking not directly at software, but at the closely related subject matter of mathematics.

Mathematics is an interesting topic to look at from a patent law perspective, because it forms one of the longest standing and perhaps least understood exceptions to patentability. Although it seems to be generally accepted that mathematics is to be excepted from patentability, the nature of the mathematics exception itself is rarely addressed. As will be seen, locating the actual source for a mathematics exception is not easy. It seems to be merely a thus-far unchallenged assumption, or perhaps an intuition. One typical explanation of the mathematics exception suggests that mathematics “[tends] to fall in the category of laws of nature, discoveries, scheme and plans, and fine arts as opposed to useful arts”.² In other words, a little bit of everything is at play. This chapter will look to both the philosophy of mathematics and the historical development of the patent system in order to determine whether a good theoretical foundation for its non-patentability can be found.

This chapter looks firstly at the historical and philosophical understandings of mathematics. It will be shown that whilst many theories as to the nature of mathematics have been advanced, no particular theory has come to dominate. This being the case, it may be wondered what the value of such a seemingly dead-end inquiry might be. This excursion into territory beyond the usual purview of the prudent lawyer, is necessary for two reasons. First, it is necessary

¹ *AT&T Corp v Excel Communications Inc* 172 F.3d 1352 (Fed Cir, 1999) at 1353.

² Rocque Reynolds and Natalie P Stoianoff, *Intellectual Property: Text and Essential Cases* (2nd ed, Federation Press, 2005) at 268, n46.

Chapter 5: The Nature of Mathematics

foundational material by which to understand the nature of innovation in the field of the mathematical arts. Whilst no one theory dominates, all theories have value in that they illuminate an understanding of the activity of mathematics, and by extension, nature of innovation in that field.

Second, it is only through an understanding of the various accounts of the nature of mathematics that the success of claims to its unpatentability can be assessed. Thus it is with this understanding that the second half of the chapter looks to the attempts of patent law to address the nature of mathematics. It will be shown how these lawyers' accounts of mathematics cannot embrace all theories of mathematics, and as a result, are unsatisfactory explanations.

1 A Mathematicians' Account of Mathematics

In order to assess the patentability of mathematics, it is important to have some understanding of what mathematics is. A starting point towards such an understanding might be to state that it is what mathematicians study, although such a definition does not advance things very far. Providing a definitive definition of mathematics is a difficult question, because it is a moving target. The nature of mathematics has changed over time, as its use has become more widespread, and as developments in mathematics combined to shake the very foundations of mathematicians' conception of their field.³

Mathematicians themselves have defined their art in broad, poetic terms. For example, it has been said that mathematics is:

- “the substance of thought writ large”;⁴
- “the queen of sciences”;⁵
- “the art of reason”;⁶ and
- “a science of patterns”.⁷

But such descriptions, although they provide some clues as to the nature of mathematics, lack the detail necessary to form a solid basis on which to build an argument either for or against patentability. A more conventional legal analysis often starts with a dictionary definition:

1 a group of related sciences, including algebra, geometry, and calculus, concerned with the study of number, quantity, shape and space and their interrelationships by using a specialized notation

³ Simon McLeish, “Mathematics” in Kenneth McLeish (ed), *Bloomsbury Guide to Human Thought*, (Bloomsbury Publishing, 1993). See also “1.1 What is Mathematics About?” in Timothy Gowers et al (eds), *The Princeton Companion to Mathematics* (Princeton University Press, 2008): “It is notoriously hard to give a satisfactory answer to the question, ‘What is mathematics?’ The approach of this book is not to try”.

⁴ Leonard Peikoff, *Objectivism: The Philosophy of Ayn Rand* (Penguin Group, 1991) at 90.

⁵ Carl Friedrich Gauss, cited in Guy W Dunnington, Jeremy Gray and Fritz-Egbert Dohse, *Carl Friedrich Gauss: titan of science* (MAA, 2004) at 44.

⁶ William P Berlinghoff, *Mathematics: The Art of Reason* (DC Heath & Company, 1968).

⁷ Michael D Resnik, *Mathematics as a science of patterns* (Clarendon Press, 1997).

- 2 mathematical operations and processes involved in the solution of a problem or study of some scientific field.⁸

From this definition the many-faceted nature of mathematics begins to reveal itself. First there is an ontological definition: that mathematics is merely a grouping term for a number of related disciplines. The definition suggests a connection amongst these disciplines, namely, the concern of all these disciplines for number, quantity, shape and space and their interrelationships. Such a definition raises a question. Is there a “unity of mathematics” which even these secondary concepts have in common? Hiding behind this definition are further questions about the nature of the objects within these fields of study – for example, what kind of objects are numbers? Are they mental constructions, or do they exist independently of us?

The second definition stands in contrast to the first in that it suggests that perhaps mathematics has no subject matter, it is merely something one *does*. By classifying mathematics as a type of science, this definition skirts around a number of epistemological questions about mathematics. In the natural sciences, knowledge is verified by empirical observation. But the content of mathematics is abstract “objects” like numbers, variables and so on. Without validation through empirical observation, how can the truth of mathematical propositions be proved? What does it mean to say that something in mathematics is “known”?⁹ It is clear that the answers to these epistemological questions may depend on the way in which the ontological questions are answered. This interdependence runs in both directions.¹⁰ It is clear then that such definitions do not provide an adequate account of the nature of mathematics by themselves.

More practically, mathematics can be described as “a process of thinking that involves building and applying abstract, logically connected networks of ideas. These ideas often arise from the need to solve problems in science, technology, and everyday life-problems...”¹¹ At the heart of mathematics lies a process of abstraction – moving the focus away from the limitations of the immediate physical problem to be solved into an abstract model.¹² This process of abstraction is based on two aspects – simplification and generalisation. For example, a simplification may be used if we need to work out the area of one face of a plank of wood, as shown in Figure 5.1 on the following page. To simplify its actual shape, one would smooth out all the bumps and variations along its edges and think of it as a rectangle, with 4 straight lines meeting at right angles. Although the resultant shape will not be perfect, the level of accuracy attained will be more than sufficient for most purposes.

A similar process might be followed where what is sought to be modelled is not one plank of wood, but many planks of wood of similar dimensions. Here, the simplification might involve using average dimensions, which might be informed by taking measurements of a small

⁸ Elspeth Summers and Andrew Holmes (eds), *Collins Australian Dictionary and Thesaurus* (3rd ed, HarperCollins Publishers, 2004).

⁹ Some of these issues are touched upon in the account of the history of mathematics below.

¹⁰ Sam Butchart, *Evidence and Explanation in Mathematics* (PhD Thesis, Monash University, 2001) at 8.

¹¹ American Association for the Advancement of Science, “The Nature of Mathematics” on *Benchmarks Online* <<http://www.project2061.org/publications/bsl/online/ch9/ch9.htm#MathematicalWorld>> (accessed 7 September 2011).

¹² This should come as no surprise in consequence of abstraction having been identified as a key aspect of software, and the isomorphism between mathematics and software discussed in Chapter 1.

Chapter 5: The Nature of Mathematics

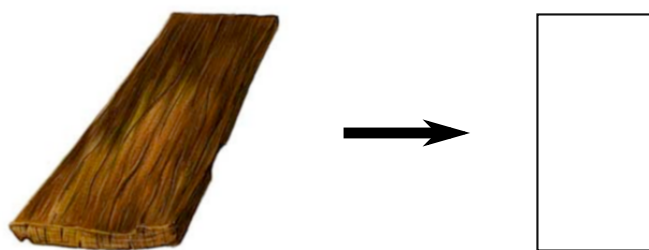


Figure 5.1: Abstraction of a real world object to a simple shape

sample of the planks involved. In such circumstances, there might be no exact correspondence between the measurements of actual pieces of wood, and the idealised average piece. But this sort of simplification makes estimations and modelling a workable task.

The advantage of such abstraction is twofold. Firstly, the process of abstraction allows mathematics to operate as “a kind of ‘place’ in which logical exactness can be ‘applied’ to the utmost degree”.¹³ Secondly, such abstract models often turn out to have applications beyond the original problem domain. For example, consider the parabola in Figure 5.2.

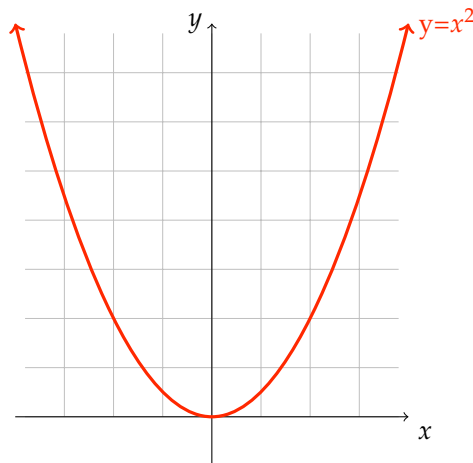


Figure 5.2: A parabola

The reflective properties of the parabola have been known for a long time, allegedly as far back as the 3rd Century BC when Archimedes used a series of reflecting mirrors arranged along a parabolic path to defend Syracuse from the Romans. The same reflective properties mean the shape is still used today in headlights and satellite dishes, the latter being shown in Figure 5.3 on the next page.¹⁴

¹³ Evandro Agazzi, “The Rise of the Foundational Research in Mathematics,” (1974) 27 *Synthese* 7 at 9.

¹⁴ Image below taken from Encyclopaedia Britannica, “Satellite communication: satellite dish antenna” *Encyclopaedia Britannica Online* <<http://www.britannica.com/eb/art-67385>> (27 December 2007).

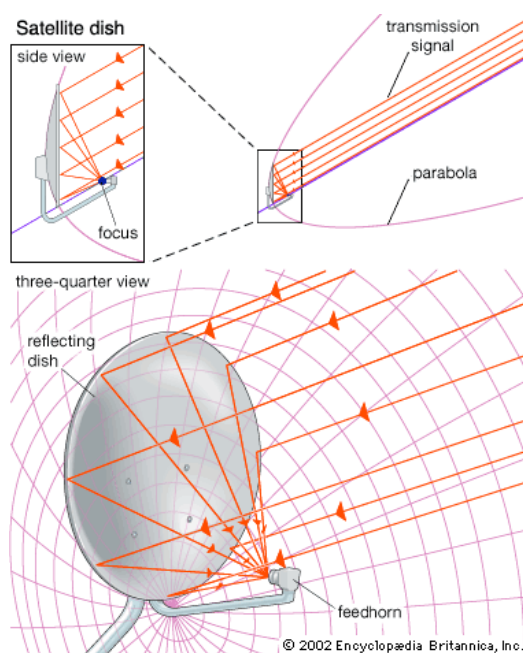
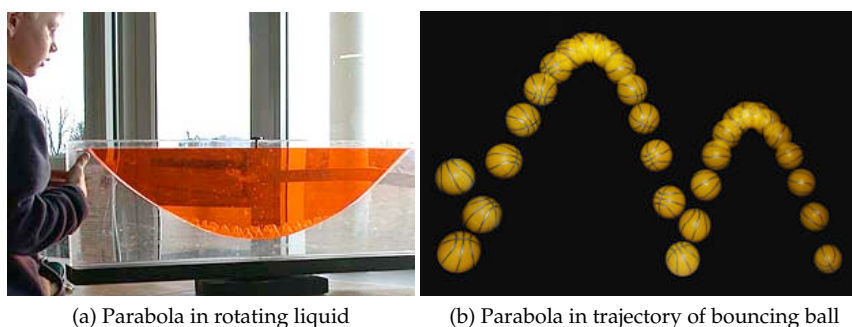


Figure 5.3: The parabolic satellite dish

But the parabola has relevance beyond its reflective properties. In the 17th Century, Galileo discovered through experimentation that the trajectory of an object travelling through the air approximately had the same shape. Isaac Newton later proved this to be the case. The same parabolic shape can also be used to model the shape of the main cable in suspension bridges, such as the Golden Gate bridge, and the “surface of a liquid confined to a container and rotated around the central axis”.¹⁵



(a) Parabola in rotating liquid

(b) Parabola in trajectory of bouncing ball

Figure 5.4: Parabolas in nature¹⁶

What this demonstrates is that not only do mathematical abstractions often owe their origins to the study of real world problems, but also that many real world problems have these deep connections to each other which can be seen via the repetitive occurrence of mathematical ‘patterns’ such as the parabola in seemingly unrelated areas.

¹⁵ Wikipedia, “Parabola” <<http://en.wikipedia.org/wiki/Parabola>> (30 October 2007).

Chapter 5: The Nature of Mathematics

The recurrence of mathematical artefacts in various physical phenomena lends an air of authenticity to the objects on which mathematics is based. Yet the study of mathematics often turns its attention to concepts which seem to have no direct relationship to the real world at all. For example, natural numbers (0, 1, 2 and so on) appear to have a concrete meaning, since they are commonly used as symbols representing our everyday experiences of collections of objects – for example 2 apples, 2 oranges, Chapter 2 and so on. Even much larger numbers such as 20,350,000,000,000 are counters for real things.¹⁷ But what about 2^{65536} ? This number exceeds the largest estimation of physically observable phenomena in the universe, namely the total number of vibrations of all subatomic particles in the universe over its entire existence.¹⁸ Even in geometry, a discipline long considered the only true mathematics due to its verifiability by human sensory experience faces similar questions. For example, a straight line may seem uncontroversial, but the straight line used in mathematics has no width, and as such cannot in reality be drawn at all. The so-called straight line one might draw on a page, or print on a printer is in reality a very thin rectangle. So given it has no manifestation in physical form, does a straight line exist?

Such questions might have been restricted to Sunday afternoon debates between mathematicians at the local pub, were it not for their practical utility. For example, negative numbers have no direct correspondence to things in the real world. You cannot hold -1 orange in your hand for example. But the use of a negative numbers has become commonplace in representing debts in accounting, on bank statements and so on. Similarly, the so-called irrational numbers such as e , $\sqrt{2}$ and π have values which cannot be represented by a simple fraction,¹⁹ yet they are necessary for geometrical calculations involving triangles and circles. Even the so-called imaginary number,²⁰ i ($\sqrt{-1}$), is used by electrical engineers to model the power of an alternating current (AC) circuit at any given time.²¹

In such a world, where the fantastical can be just as useful as the real, it should come as no surprise that it can be difficult to distinguish truth from fantasy. It is therefore important to have some sort of mechanism for proving the truth value of mathematical propositions. The search for a solid foundation for mathematical truths is as much a question of the history of

¹⁶ Diagrams taken from Wikipedia, Parabola <<http://en.wikipedia.org/wiki/Parabola>> (30 October 2007)

¹⁷ This number represents the upper bounds of how many bytes of email were sent in the year 2000. Taken from Peter Lyman and Hal R Varian, "How Much Info? Internet" (School of Information Management and Systems, University of California, 2000) <<http://www2.sims.berkeley.edu/research/projects/how-much-info/internet.html>> (7 September 2011).

¹⁸ David Isles, "What Evidence is there that 2^{65536} is a natural number?" (1992) 33(4) *Notre Dame Journal of Formal Logic* 465.

¹⁹ It is said that when the Greek mathematician Hipassus reported his discovery of the irrationality of $\sqrt{2}$ he "became so much hated by the Pythagoreans, that not only they cast him out of the community; they built a shrine for him as if he were dead". Hyamblicus of Chalkis, *De vita pythagoras*, at 246-247. Cited in Michael Lahanas, "Irrational numbers" <<http://www.mlahanas.de/Greeks/Irrational.htm>> (28 December 2007). Alternative accounts suggest he was thrown overboard from the ship he was on at the time and drowned. See Richard Mansfield, "Real and Complex Numbers" in *Logic: The Bridge to Higher Math* <<http://www.math.psu.edu/melvin/logic/node8.html>> (7 September 2011).

²⁰ This derogatory term was coined by Descartes, who rejected their use because "while we can always conceive [of the square roots of negative numbers], still there is ... no quantity corresponding to those we conceive." See Rene Descartes, *Discourse on Method, Optics, Geometry, and Meteorology* (PJ Olscamp (trans), Hackett Publishing, 2001) at 236.

²¹ See for example, John A Masters, "Imaginary Numbers Really Exist! (I'm Really, Really Serious!)" on *Least Significant Bits* <<http://jamspeaks.blogspot.com/2007/09/imaginary-numbers-really-exist-im.html>> (7 September 2011).

mathematics as a question of philosophy. And it is difficult terrain to traverse. There are nearly as many views on the nature of mathematics as there are mathematicians. And no one school of thought has become dominant, despite the considerable debate on the issue.

For present purposes, it will be sufficient to look at a representative sample, as most alternative theories can be considered to be derivatives, mixtures or variations of these basic theories. A final resolution of this debate within the pages of this thesis will not be attempted. An understanding of these theories is however central to this thesis, because the school of thought to which one subscribes greatly influences the reasoning by which it might be argued that mathematics is non-patentable. If it is asserted that mathematical objects and principles exist *a priori*, then mathematics is non-patentable because it amounts to a discovery. If one thinks they are merely mental constructions, then mathematics is non-patentable as ideas. If you take what might be called a inclusive view, as this author does, that no one theory of mathematics will provide the ultimate guidance but all have something useful to say, then the approach to reconciliation lies down a different path. That point will be revisited at the end of this chapter. Before adopting that course however, it is necessary to embark upon a discussion of the philosophy of mathematics. To fully understand the philosophy of mathematics it is necessary to have some regard to the history of mathematics. A complete history of mathematics, in all its glorious detail, lies well beyond the scope of this thesis.²² Instead, only a brief history is attempted, in order that some context might be given to the various philosophies of mathematics which have been put forward, and to tie the development of mathematics to the various milestones of the patent system set out in Chapter 2.

The history and philosophies of mathematics

The origins of mathematics lies in the use of numbers, which on a conservative estimate pre-dates the birth of the patent system by at least 3,000 years.²³ At this earliest stage, “the crucial point that marks the emergence of mathematics from mere counting is that of abstraction”.²⁴ But it was not until the Ancient Greek civilisation that “mathematics would make the transition from a strictly practical discipline to an intellectual one, where knowledge for its own sake became the main goal.”²⁵

²² In any event, such histories have already been compiled. See for example, Morris Kline, *Mathematical Thought from Ancient to Modern Times* (Oxford University Press, 1972); David Eugene Smith, *History of Mathematics* (Courier Dover Publications, 1958); Walter William Rouse Ball, *A Short Account of the History of Mathematics*, (4th ed, Project Gutenberg, 2010); Florian Cajori, *A History of Mathematics* (Project Gutenberg, 2010). For a more concise account see Dirk J Struik, *A Concise History of Mathematics*, (3rd ed, Dover Publications, 1967); Robert Tubbs, *What is a Number? Mathematical Concepts and Their Origins* (John Hopkins University Press, 2009); Eli Maor, *To Infinity and Beyond: A Cultural History of the Infinite* (Birkhäuser, 1987).

²³ The earliest source of information on the mathematics of the ancient Egyptians comes from the Moscow Mathematical Papyrus which dates from around 1850 B.C.. See WS Anglin, *Mathematics, a concise history and philosophy* (Springer Verlag, 1994) at 1-2. Struik places the origins of mathematics at the time of the “transition ... from the mere gathering of food to its actual production, from hunting and fishing to agriculture.”: Dirk J Struik, *A Concise History of Mathematics*, (3rd ed, Dover Publications, 1967) at 7, which he puts at “perhaps ten thousand years ago”.

²⁴ Simon McLeish, “Mathematics” in Kenneth McLeish (ed), *Bloomsbury Guide to Human Thought* (Bloomsbury Publishing, 1993).

²⁵ Eli Maor, *To Infinity and Beyond: A Cultural History of the Infinite* (Birkhäuser, 1987) at 31.

Chapter 5: The Nature of Mathematics

Ancient times

For nearly 2000 years, mathematics was dominated, in both theoretical and practical terms, by the Ancient Greeks,²⁶ whose “characteristic achievements [were] the development of rigorous deductive proofs and the geometrical representation of the cosmos for astronomical and geographical purposes.”²⁷ In around 375BC, Plato wrote *The Republic*, in which he posited a theory of knowledge in which the realm of the intelligible was to be given more weight than the realm of the visible.²⁸ The visible world was in a constant state of flux, and was capable of founding only opinion or belief. The “real” world, and true basis for truth or knowledge was thus the intelligible world, where the immutable ideal forms (being archetypes or abstractions of visible-world objects) could found similarly immutable truths.²⁹ As such, mathematics was both separate and superior to empirical sciences. Plato divided the subject matter of mathematics into number theory (arithmetic), geometry, astronomy and harmonics (or music). These subjects made up the *Quadrivium*, the second stage of his seven liberal arts, which were to be mastered “before turning to dialectic and the ascent to the Good”.³⁰ Within the *Quadrivium*, the subjects were paired, geometry with astronomy, and arithmetic with music, each pair corresponding to the abstract and concrete aspects of an area.³¹

In relation to practical mathematics, it was around 300BC that Euclid systematized geometry³²

²⁶ “To be sure, mathematics as a science had already reached quite an advanced stage before the Greek era... But the ancient mathematics of the Hindus, the Chinese, the Babylonians and the Egyptians confined itself solely to practical problems of daily life, such as the measurement of area, volume, weight and time”: Eli Maor, *To Infinity and Beyond: A Cultural History of the Infinite* (Birkhäuser, 1987) at 2-3.

²⁷ Ian Mueller, ‘Earlier Greek Mathematics’ in Donald J Zeyl (ed), *Encyclopedia of Classical Philosophy* (1997).

²⁸ See Plato, “The Republic” in *The Dialogues of Plato translated into English with Analyses and Introductions by B. Jowett, M.A. in Five Volumes* (3rd ed, Oxford University Press, 1892) at line 511 (Stephanus numbering): “[c]orresponding to these four divisions [achieved by splitting the realms of the visible and intelligible into two], let there be four faculties in the soul—reason answering to the highest, understanding to the second, faith (or conviction) to the third, and perception of shadows to the last—and let there be a scale of them, and let us suppose that the several faculties have clearness in the same degree that their objects have truth.”

²⁹ “[T]he principal doctrines of the dialogues express Plato’s own philosophy: There is a realm of eternal and changeless objects called “Ideas” or “Forms”; these are the most real entities and the basic objects of knowledge; the Form of the Good is the highest object of understanding; the observable world is a deficient reflection of the Forms...”: Richard Kraut, “Plato (427-347 B.C.E.)” in Donald J Zeyl, Daniel Devereux and Phillip Mitsis (eds), *Encyclopedia of Classical Philosophy* (Greenwood Press, 1997) at 390. In the *Phaedo*, Plato gives an example based on the notion of equality demonstrates the existence of the Forms. Kraut (*ibid* at 397) summarises it as follows:

Equal sticks are as much an example of equality as inequality, since each member of the pair is also unequal to other sticks; but the Form is precisely what equality is and can never be characterized as unequal. This is why someone who knows what equality is must have acquired this knowledge by looking to the Form and not merely to objects whose equality is detected by the senses.

³⁰ Ian Mueller, “Earlier Greek Mathematics” in Donald J Zeyl, Daniel Devereux and Phillip Mitsis (eds), *Encyclopedia of Classical Philosophy* (Greenwood Press, 1997) at 316, citing Plato’s *Republic*.

³¹ Luigi Borzacchini, “2. Greek mathematics and Pythagoras” in *Being and Sign II: Axiomatic deductive method and infinite in Greek mathematics*, (1995) <<http://www.dm.uniba.it/~psiche/bas2/node3.html>> (2 November 2010).

³² It may be noticed that geometry seems to have preoccupied the ancient Greeks to a much greater extent than arithmetic. For example, it is suggested that geometry was “the most philosophically satisfying branch of mathematics by the ancient Greeks; and indeed until the end of the Middle Ages, the word ‘mathematics’ meant geometry”: Simon McLeish, “Mathematics” in Kenneth McLeish (ed), *Bloomsbury Guide to Human Thought* (Bloomsbury Publishing, 1993). Plato considered geometry to be “the clearest example of an human natural access to the Forms world”: Luigi Borzacchini, “2. Greek mathematics and Pythagoras” in *Being and Sign II: Axiomatic deductive method and infinite in Greek mathematics*, (1995) <<http://www.dm.uniba.it/~psiche/bas2/node3.html>> (2 November 2010). This may in part be due to the limitations imposed on number theory by the Pythagoreans, whose theory of numbers extended only to integers (or whole numbers). As such, many problems involving non-integers (for example, so-called irrational numbers

in *The Thirteen Books of the Elements*. Euclid's system was based upon five axioms, the truth of which was said to be self-evident. From these five axioms, "a vast amount of knowledge [was] derived in steps which are apparent to any reasonably intelligent and sufficiently well-trained reader."³³ However,

its dependence on visual intuition (whose consequent deductive gaps were already noted by Archimedes), together with the challenge of Euclid's infamous fifth postulate (about parallel lines), and the famous unsolved problems of compass and straightedge construction, established an agenda for generations of mathematicians.³⁴

Despite these limitations "[t]he *Elements* form, next to the Bible, probably the most reproduced and studied book in the history of the Western world."³⁵

Platonism Plato's theory of the Forms gives rise to the most long-lived of the philosophies of mathematics, platonism. On a platonist account, the realm of mathematics is populated with "abstract, necessarily existing objects, independent of the human mind".³⁶ These mathematical objects exist in the realm of the intelligible, rather than the visible.³⁷ Despite the name however, platonism is not limited to the terms of Plato's account, and is normally "defined and debated independently of its original historical inspiration".³⁸

Platonist accounts have the advantage in that they support an objective notion of proof which puts mathematics on a similar basis to physics. As a consequence, "[t]he statements of mathematics are true or false depending on the properties of those [abstract mathematical] entities, independent of our ability, or lack thereof, to determine which."³⁹ It is perhaps for this reason that Platonist accounts continue to enjoy a popularity with mathematicians, at least as a starting point,⁴⁰ because the "Platonist attitude to objects of investigation is inevitable for a mathematician: during his everyday work he is used to treat numbers, points, lines etc. as the 'last reality', as a specific independent 'world'. This sort of platonism is an essential aspect of mathematical method, the source of the surprising efficiency of mathematics in the natural

such as π , or $\sqrt{2}$ were "thus resolved ... in a geometric manner instead": Wilbur R Knorr, "Transcript of a Lecture Delivered at The Annual Convention of the History of Science Society, Atlanta, December 28, 1974" in Jean Christianidis (ed) *Classics in the history of Greek mathematics* (Kluwer Academic Publishers, 2004).

³³ Simon McLeish, "Mathematics" in Kenneth McLeish (ed), *Bloomsbury Guide to Human Thought* (Bloomsbury Publishing, 1993).

³⁴ CJP, "Philosophy of Mathematics" in Robert Audi (ed), *The Cambridge Dictionary of Philosophy* (2nd ed, Cambridge University Press, 1999) at 681.

³⁵ Dirk J Struik, *A Concise History of Mathematics* (3rd ed, Dover Publications, 1967) at 50.

³⁶ WS Anglin, *Mathematics, a concise history and philosophy* (Springer Verlag, 1994) at 218.

³⁷ See page 194 above.

³⁸ Øystein Linnebo, "Platonism in the Philosophy of Mathematics," *Stanford Encyclopedia of Philosophy*, 18 Jul 2009 <<http://plato.stanford.edu/entries/platonism-mathematics/>> (22 November 2010).

³⁹ Penelope Maddy, *Realism in Mathematics* (Clarendon Press, 1990) at 21.

⁴⁰ "In everyday life, we speak as Platonists, treating the objects of our study as real things that exist independently of human thought. If challenged on this, however, we retreat to some sort of formalism, arguing that in fact we are just pushing symbols around without making any metaphysical claims. Most of all, however, we want to do mathematics rather than argue about what it actually is. We're content to leave that to the philosophers." Jean Dieudonné. Cited in Fernando Gouvea, "Book Review: What is Mathematics Really? by Reuben Hersh," *MAA Online*, <<http://test.maa.org/reviews/whatis.html>> (9 September 2011).

Chapter 5: The Nature of Mathematics

sciences and technology”.⁴¹ Indeed some “working realists” seek to avoid the philosophical debate entirely, by asserting a “methodological view that mathematics should be practiced *as if* platonism was true”.⁴²

This practical convenience comes at a price however, since this other-worldliness raises questions, namely “[w]here is this world and how do we make contact with it? How is it possible for our mind to have an interaction with the Platonic realm so that our brain state is altered by that experience?”⁴³ In other words, accepting the Platonist view of mathematical entities requires a leap of faith:

Change “mathematics” to “God” and little else might seem to change. The problem of human contact with some spiritual realm, of timelessness, of our inability to capture all with language and symbol – all have their counterparts in the quest for the nature of Platonic mathematics.⁴⁴

It is perhaps for this reason that modern interpretations of the Platonist view abandon the epistemological claims as to how to access the world of Forms might occur through some sort of mystical mental faculty. Brown for example points out that ordinary perception of the physical world may be understood as a matter of physiology, but how those perceptions become sensations and beliefs “is a very great mystery. It is just as great a mystery as how mathematical entities bring about mathematical beliefs.”⁴⁵ But for the discipline of mathematics, with its “virtually unparalleled standard of intellectual rigor and exactitude”⁴⁶ this explanation remains a little unsatisfying.

15th Century

As noted in Chapter 2, a patent system was set up in Venice in the 15th Century.⁴⁷ At this time, the theoretical province of mathematics was still dominated by the *Quadrivium*, which was taught at medieval universities.⁴⁸ Over a similar period, the “Italians, and particularly

⁴¹ Karlis Podnieks, “Platonism, Intuitionism and the Philosophy of Mathematics” in (1992) 31 *Semiotika i informatika* 150, English translation available at <<http://www.ltn.lv/~podnieks/gt1.html>> (29 Oct 2007).

⁴² Øyestein Linnebo, “Platonism in the Philosophy of Mathematics” *Stanford Encyclopedia of Philosophy*, 18 July 2009 <<http://plato.stanford.edu/entries/platonism-mathematics>> (24 November 2010). Linnebo cites Bernays and Shapiro as adherents of this view: Paul Bernays, “On Platonism in Mathematics” in Paul Benacerraf and Hilary Putnam (eds) *Philosophy of Mathematics: Selected Readings* (2nd ed, Cambridge University Press, 1983); Stewart Shapiro, *Philosophy of Mathematics: Structure and Ontology* (Oxford University Press, 1997) at 21-27 and 38-44.

⁴³ John D Barrow, *Pi in the Sky: Counting, Thinking, and Being* (Clarendon Press, 1992) at 272.

⁴⁴ John D Barrow, *Pi in the Sky: Counting, Thinking, and Being* (Clarendon Press, 1992) at 272.

⁴⁵ James Robert Brown, *Philosophy of Mathematics: An Introduction to the World of Proofs and Pictures* (Routledge, 1999) at 15.

⁴⁶ Michael J White, “Plato and Mathematics,” in Hugh H Benson (ed), *A Companion to Plato* (Blackwell Publishing, 2005) at 228.

⁴⁷ See 2 on page 40.

⁴⁸ “The early and central Middle Ages had inherited from antiquity the scheme of the seven Liberal Arts: grammar, rhetoric, dialectics (... the *trivium*), and the four mathematical arts, arithmetic, geometry, music and astronomy (the *quadrivium*).”: Jens Høyrup, *In measure, number, and weight: studies in mathematics and culture* (SUNY Press, 1994) at 177. “The quadrivium functioned as the source of theoretical and exact science for medieval university students.”: Edward Grant, *The foundations of modern science in the Middle Ages: their religious, institutional and intellectual contexts* (1996) at 44. See also Frank Swetz, *Capitalism and Arithmetic: The New Math of the 15th Century* (David E Smith (trans), Open Court Publishing, 1987) at 14, noting the lack of prominence given to arithmetic.

the Venetians, realized the importance of the use of arithmetic in their daily business transactions.”⁴⁹ As such, the Italians “not only adapted new mathematical techniques for their commercial interests, but also became innovators – the practice of double-entry bookkeeping originated in northern Italy during this period.”⁵⁰ As such, “the climate of mathematics and mathematics education in Italy was of such excellence that it attracted students from all over Europe.”⁵¹ Such mathematical education was not necessarily the province of universities, but developed through the emergence of “reckoning schools” through which would-be merchants learned the mathematical skills which they would need to begin their apprenticeships. “The existence of these *scuola d’abbaco*, schools of the abacus, was testimony to the importance of mathematics and computational proficiency in the business world of the time.”⁵² Swetz and Smith also conclude from this popular interest in mathematics as a useful discipline, that “[m]athematics was moving from the realm of scholastic speculation to the applications of manufacture and the marketplace.”⁵³ Incremental advances in computational methods saw the gradual emergence of improved notation systems which would eventually lead to the emergence of algebra in its modern form.⁵⁴

Also at this time, the development of geometry was being driven by Renaissance artists in a number of ways “for translating the reality of 3-dimensional natural phenomena onto 2-dimensional surfaces, producing virtually realistic copies”⁵⁵ (perspective) and also “in determining the correct proportions for the figures they drew”⁵⁶ (proportion). Leonardo da Vinci is perhaps the best known exponent, but is by no means the only example.⁵⁷

Also contributing to the development of mathematics over this period was “the productive use and further perfection of machines”⁵⁸ which led to “theoretical mechanics and the scientific study of motion and of change in general”.⁵⁹

17th Century

The traditional limitations on mathematics, requiring a close correlation with the real world, began to be questioned in the 17th Century when mathematicians such as Descartes began to

⁴⁹ Frank Swetz, *Capitalism and Arithmetic: The New Math of the 15th Century* (David E Smith (trans), Open Court Publishing, 1987) at 11.

⁵⁰ Frank Swetz, *Capitalism and Arithmetic: The New Math of the 15th Century* (David E Smith (trans), Open Court Publishing, 1987) at 12.

⁵¹ Frank Swetz, *Capitalism and Arithmetic: The New Math of the 15th Century* (David E Smith (trans), Open Court Publishing, 1987) at 13.

⁵² Frank Swetz, *Capitalism and Arithmetic: The New Math of the 15th Century* (David E Smith (trans), Open Court Publishing, 1987) at 13.

⁵³ Frank Swetz, *Capitalism and Arithmetic: The New Math of the 15th Century* (David E Smith (trans), Open Court Publishing, 1987) at 34.

⁵⁴ Dirk J Struik, *A Concise History of Mathematics*, (3rd ed, Dover Publications, 1967) at 93-97.

⁵⁵ Joseph W Dauben, “Galileo and Perspective: The Art of Renaissance Science” *American Physical Society*, 2002, <<http://www.aps.org/publications/apsnews/200201/backpage.cfm>> (21 December 2007).

⁵⁶ Joseph W Dauben, “Galileo and Perspective: The Art of Renaissance Science” *American Physical Society*, 2002, <<http://www.aps.org/publications/apsnews/200201/backpage.cfm>> (21 December 2007).

⁵⁷ See Robert Tubbs, *What is a Number? Mathematical Concepts and Their Origins* (John Hopkins University Press, 2009) at 141-151 who notes Alhazen’s theory of vision, as well as the work of di Bondone and Brunelleschi, in addition to da Vinci. See also Dirk J Struik, *A Concise History of Mathematics* (3rd ed, Dover Publications, 1967) at 90.

⁵⁸ Dirk J Struik, *A Concise History of Mathematics* (3rd ed, Dover Publications, 1967) at 99.

⁵⁹ Dirk J Struik, *A Concise History of Mathematics* (3rd ed, Dover Publications, 1967) at 99.

Chapter 5: The Nature of Mathematics

approach geometrical problems using the tools of algebra.⁶⁰ So when the *Statute of Monopolies* was passed at the beginning of the 17th Century, mathematics was in a state of flux. In this period, “the great philosophers [were] also the great mathematicians, and the great problems of mathematics [were] great problems of philosophy.”⁶¹ For mathematicians, the century was dominated by issues such as “the relative merits of geometry versus arithmetic and algebra centred on questions of the ability of the human mind to reason on symbols, the legitimacy of the negative and imaginary numbers, and finally the ultimate source of the numbers of arithmetic.”⁶² These issues illustrate the concern amongst mathematicians and philosophers about the increasingly abstract nature of the subject matter with which mathematics was concerned.

Debates as to the nature of mathematics at this time were dominated by two schools of thought, namely rationalism, and empiricism. Rationalism, or continental rationalism as it is sometimes called in this context, was supported largely by algebraists such as Descartes and Leibniz. Empiricism was championed by (largely English) geometers, such as Locke, Berkeley, Hume and J.S. Mill.

Rationalism The rationalist account, notably championed by Descartes, Spinoza and Leibniz, can be considered similar in many respects to platonism,⁶³ although it does not require any independent existence of mathematical objects.⁶⁴ Rationalists posit that both the source and ultimate arbiter of knowledge is human reason. As a tool of reason, mathematics is afforded a special significance within rationalist philosophy. For example, Descartes suggested that reason “is a more powerful instrument of knowledge than any other that has been bequeathed to us by human agency, as being the source of all others.”⁶⁵

Empiricism Empiricism, usually set in contrast to rationalism, asserts that mathematics ultimately depends for its validation on evidence that is observable by the senses. That is, mathematical truths are verified by empirical research, just as in any other science. This approach is underscored by a

⁶⁰ Helena M Pycior, “Mathematics and Philosophy: Wallis, Hobbes, Barrow, and Berkeley” (1987) 48(2) *Journal of the History of Ideas* 265 at 267. See also Dirk J Struik, *A Concise History of Mathematics* (3rd ed, Dover Publications, 1967) at 102-104.

⁶¹ William B Ewald, *From Kant to Hilbert* (Oxford University Press, 1999) Volume 1 at 1.

⁶² Helena M Pycior, “Mathematics and Philosophy: Wallis, Hobbes, Barrow, and Berkeley” (1987) 48(2) *Journal of the History of Ideas* 265 at 267.

⁶³ This is so in that rationalist accounts maintain a distinction between the visible and intelligible realms, and place an emphasis on the intelligible as superior.

⁶⁴ “[A] signature doctrine of rationalism is the doctrine of innate ideas, according to which the mind has built into it not just the structure of knowledge but even its content”: Thomas M Lennon and Shannon Dea, “Continental Rationalism” *Stanford Encyclopedia of Philosophy*, 21 November 2007 <<http://plato.stanford.edu/entries/continental-rationalism/>> (27 Sep 2010). However, note that Descartes’ theory as to the nature of mathematical objects (René Descartes, “Meditation V: On the Essence of Material Objects and More on God’s Existence” in *Meditations on First Philosophy* (Areté Press, 1986) at 35):

Suppose, for example, that I have a mental image of a triangle. While it may be that no figure of this sort does exist or ever has existed outside my thought, *the figure has a fixed nature (essence or form), immutable and eternal, which hasn’t been produced by me and isn’t dependent of my mind.* (emphasis added)

⁶⁵ René Descartes, “Rules for the Direction of the Mind” in *Key Philosophical Writings* (Wordsworth Editions, 1997) at 13.

view that the world is largely, or even, entirely a product of chance. On the empiricist account, the universe consists of many independent individuals, which, if they are connected, are so only accidentally, reducing causation to nothing more than a matter of constant conjunction. ... Under such circumstances, only experience of the world can provide knowledge of it.⁶⁶

John Locke,⁶⁷ George Berkeley,⁶⁸ and David Hume⁶⁹ were the primary exponents of empiricism, holding that all knowledge is derived from sensory experience.⁷⁰ J.S. Mill was also an important early supporter of the notion of mathematics as an empirical science.⁷¹ Mill argued for example that the laws of arithmetic were nothing more than “inductive generalisations from observed facts”.⁷²

What both empiricist and rationalist accounts have in common is the assertion of a harmony between mathematics and the real world, and in particular, physics. Whilst the Greeks had maintained a sharp distinction between it and science, the mathematics of the time of Descartes and Newton was not so distinct from physics. Kline notes that:

as the province of mathematics expanded, and mathematicians not only relied upon physical meanings to understand their concept but accepted mathematical argu-

⁶⁶ Thomas M Lennon and Shannon Dea, “Continental Rationalism” *Stanford Encyclopedia of Philosophy*, 21 November 2007 <<http://plato.stanford.edu/entries/continental-rationalism/>> (27 Sep 2010).

⁶⁷ John Locke, *An Essay Concerning Human Understanding*, (John W Yolton (ed), Dent, 1976). According to Locke, the mind at birth “is a *tabula rasa* or blank sheet until experience in the form of sensation and reflection provide the basic materials — simple ideas — out of which most of our more complex knowledge is constructed.”: William Uzgalis, “John Locke,” *Stanford Encyclopedia of Philosophy*, 2 September 2001, <<http://plato.stanford.edu/entries/locke/>> (30 November 2010). For Locke, mathematics was a “true method of advancing knowledge ... by considering our abstract ideas... [M]athematicians, ... from very plain and easy beginnings, by gentle degrees, and a continued chain of reasonings, proceed to the discovery and demonstration of truths that appear at first sight beyond human capacity.”: Locke, *An Essay Concerning Human Understanding*, Book IV, Chapter 12, paragraph 7.

⁶⁸ George Berkeley, *Treatise Concerning the Principles of Human Knowledge* (Project Gutenberg, 2009) <<http://www.gutenberg.org/files/4723/4723-h/4723-h.htm>> (29 November 2010). Unlike Locke, Berkeley attributed to mathematics a merely secondary role, noting that “how celebrated soever [mathematics] may be for [its] clearness and certainty of demonstration, which is hardly anywhere else to be found, [it] cannot nevertheless be supposed altogether free from mistakes”: at [108].

⁶⁹ David Hume, *An Enquiry Concerning Human Understanding* (2006, Project Gutenberg) <<http://www.gutenberg.org/dirs/etext06/8echu10h.htm>> (29 November 2010). Similarly to Berkeley, Hume seems satisfied with mathematics to the extent that it maintains a connection with the physical world, but notes that “the finer sentiments of the mind, the operations of the understanding, the various agitations of the passions, though really in themselves distinct, easily escape us, when surveyed by reflection; nor is it in our power to recall the original object, as often as we have occasion to contemplate it. Ambiguity, by this means, is gradually introduced into our reasonings: Similar objects are readily taken to be the same: And the conclusion becomes at last very wide of the premises”: at [48].

⁷⁰ Mention should be made at this point of Aristotle, whose philosophy “contains seeds of empiricism”: Stewart Shapiro, *Thinking About Mathematics* (Oxford University Press, 2000) at 62. Aristotle agreed with Plato as to the existence of Forms (or universals, as he called them). However, Aristotle contended that these forms “exist in perceptible objects”: Aristotle, *Metaphysics*, Book M, (Annas (trans), Clarendon Press, 1976), cited in Stewart Shapiro, *Thinking About Mathematics* (Oxford University Press, 2000) at 64. These perfect forms are accessed through a faculty of abstraction, in which physical objects are contemplated, and some of their features are abstracted away. See Stewart Shapiro, *Thinking About Mathematics* (Oxford University Press, 2000) at 66.

⁷¹ See John Stuart Mill, *A System of Logic Ratiocinative and Inductive being a Connected View of the Principles of Evidence and the Methods of Scientific Investigation*, (8th ed, Harper & Brothers, 1882) <<http://www.gutenberg.org/files/27942/27942-pdf.pdf>> (10 September 2011), and in particular Chapters 5 and 6 of Book II, Chapter 24 of Book III. For a summary of Mill’s position, see Donald Gillies, “An Empiricist Philosophy of Mathematics and Its Implications for the History of Mathematics,” in Emily Grosholz and Herbert Breger (eds), *The Growth of Mathematical Knowledge* (Springer, 2000) at 41-57.

⁷² Donald Gillies, “An Empiricist Philosophy of Mathematics and Its Implications for the History of Mathematics,” in Emily Grosholz and Herbert Breger (eds), *The Growth of Mathematical Knowledge* (Springer, 2000) at 41.

Chapter 5: The Nature of Mathematics

ments because they gave sound physical results, the boundary between mathematics and science became blurred.

... In fact, it would be difficult to name an outstanding mathematician of the [18th] century who did not take a keen interest in science. As a consequence these men did not wish or seek to make any distinctions between the two fields.⁷³

Transcendental Idealism Kant's transcendental idealism stands halfway between rationalist and empiricist accounts.⁷⁴ Rather than the typical empiricist account by which sensory perceptions were passively received into the mind, Kant held that "the mind of the knower makes an active contribution to experience of objects before us".⁷⁵ For Kant, "[a]pppearances are not things in themselves. Empirical intuition is possible only through the pure intuition (of space and time)".⁷⁶ It is by reference to this same "pure intuition" that Kant believed mathematical concepts were constructed, giving them the same verifiability as sensory information, and thereby differentiating mathematics from other intellectual pursuits such as philosophy. Kant's theory was "developed ... in the context of the actual mathematical practices of his predecessors and contemporaries, and he produced thereby a coherent and compelling account of early modern mathematics."⁷⁷ As a result, Kant's theory was to dominate until the upheaval of mathematics in the second half of the 19th Century.

So at the time that the modern patent system was being forged, the debate between mathematicians as to the ultimate source of truth was in hot dispute. As will be seen, it could perhaps be argued that this debate has never been resolved, with many modern philosophies being more refined varieties of rationalism or empiricism. But it is worth mentioning at this point, that it seems to be acknowledged, or at least inferred, that the doing of mathematics, even on an empiricist view, is an activity set apart from the physical world. On either view, mathematics is a mental activity which is true by virtue of its reliance on reason, or verified after the fact through empirical observation of the physical world.

19th Century

By the 19th Century, the rationalist/empiricist debate had been largely resolved in favour of empiricism.⁷⁸ However, by the second half of the century, the growing abstractness and complexity of mathematics was causing the link between the external world, and the study of math-

⁷³ Morris Kline, *Mathematical Thought from Ancient to Modern Times* (Oxford University Press, 1972) at 395.

⁷⁴ What follows is an inevitably oversimplified, broad brush account of Kant's philosophy of mathematics. For a more detailed account see Lisa Shabel, "Kant's Philosophy of Mathematics" in Paul Guyer (ed) *The Cambridge Companion to Kant and Modern Philosophy* (Cambridge University Press, 2006).

⁷⁵ Matt McCormick, "Immanuel Kant: Metaphysics" *Internet Encyclopedia of Philosophy*, 30 June 2005 <<http://www.iep.utm.edu/kantmeta/>> (7 December 2010).

⁷⁶ Immanuel Kant, *Critique of Pure Reason* (Paul Guyer, Allen W Woods (trans, eds), Cambridge University Press, 1998) at 289 (A165/B206).

⁷⁷ Lisa Shabel, "Kant's Philosophy of Mathematics" in Paul Guyer (ed) *The Cambridge Companion to Kant and Modern Philosophy* (2006) at 119.

⁷⁸ Leon Horsten, "Philosophy of Mathematics" *Stanford Encyclopedia of Philosophy*, 25 September 2007 <<http://plato.stanford.edu/entries/philosophy-mathematics>> (10 September 2011)

ematics to be questioned. In particular, Mill's views were harshly satirised by Gottlob Frege, the founder of the logicist school,⁷⁹ who remarked:

What, then, are we to say of those who, instead of advancing this work where it is not yet completed, despise it, and betake themselves to the nursery, ... there to discover, like John Stuart Mill, some gingerbread or pebble arithmetic!⁸⁰

Frege pointed to our understanding of operations on large numbers to illustrate the limitations of Mill's philosophy. He pointed out that "[o]n Mill's view we could actually not put $1,000,000 = 999,999 + 1$ unless we had observed a collection of things split up in precisely this peculiar way."⁸¹

Over the nineteenth century, "as in the two preceding centuries ... the progress in mathematics brought with it larger changes barely perceptible in the year-to-year developments but vital in themselves and in their effect on future developments."⁸² Because of the "vast expansion in subject matter and in the opening of new fields as well as the extension of older ones",⁸³ "by 1870 mathematics had grown into an enormous and unwieldy structure, divided into a large number of fields in which only specialists knew the way."⁸⁴ A desire to synthesize these various fields lead to the a new set of unifying principles centred around group theory, which has been described as "a supreme example of the art of mathematical abstraction".⁸⁵ Over the preceding period, the work of Gauss, Lobachevsky, Bolyai and Riemann had also challenged the supremacy of Euclidean geometry by demonstrating that it was possible to construct an alternative geometry in which Euclid's fifth postulate did not hold.⁸⁶ The development of non-Euclidean geometries was "from the standpoint of intellectual importance and ultimate effect on the nature of mathematics, the most consequential development [in mathematics]."⁸⁷ That such geometries could be internally consistent, suggested that

the set of axioms one chooses as the foundation of a mathematical structure [such as geometry] is, to a certain extent, arbitrary; change one or more of the axioms, and a different structure will emerge. Whether the new structure agrees with the 'real' physical world is totally irrelevant; what matters is logical consistency alone.⁸⁸

One result of this shaking of the foundations of geometry was that it fell "into disfavour because mathematicians found that they had unconsciously accepted facts on an intuitive ba-

⁷⁹ The logicist philosophy of mathematics is considered below.

⁸⁰ Gottlob Frege, *The Basic Laws of Arithmetic* (Montgomery Furth (trans), University of California, 1967) at vii.

⁸¹ Gottlob Frege, *The Basic Laws of Arithmetic* (Montgomery Furth (trans), University of California, 1967) at 10-11.

⁸² Morris Kline, *Mathematical Thought from Ancient to Modern Times* (Oxford University Press, 1972) at 1023.

⁸³ Morris Kline, *Mathematical Thought from Ancient to Modern Times* (Oxford University Press, 1972) at 1023.

⁸⁴ Dirk J Struik, *A Concise History of Mathematics*, (3rd ed, Dover Publications, 1967) at 174. See also Morris Kline, *Mathematical Thought from Ancient to Modern Times* (Oxford University Press, 1972) at 1024.

⁸⁵ James R Newman, *The World of Mathematics, Volume 3* (Courier Dove Publications, 2000) at 1534.

⁸⁶ Dirk J Struik, *A Concise History of Mathematics*, (3rd ed, Dover Publications, 1967) at 167.

⁸⁷ Morris Kline, *Mathematical Thought from Ancient to Modern Times* (Oxford University Press, 1972) at 1023. The impact of non-Euclidean geometries was however greatly magnified by the introduction of more and more concepts with little physical relevance into the domain of mathematics, from irrational to negative and complex numbers, "quaternions, ... complex elements in geometry, n -dimensional geometry, bizarre functions, and transfinite numbers": at 1029.

⁸⁸ Eli Maor, *To Infinity and Beyond: A Cultural History of the Infinite* (Birkhäuser, 1987) at 124.

Chapter 5: The Nature of Mathematics

sis, and their supposed proofs were consequentially incomplete. The danger that this would continually recur made them believe that the only sound basis for geometry would be arithmetic.”⁸⁹ However, “even arithmetic and the analysis built on it soon became suspect. The creation of non-commutative algebras ... raised the question of how one can be sure that ordinary numbers possess the privileged property of truth about the real world.”⁹⁰ The result was that “[b]y 1900 mathematics had broken away from reality; it had clearly and irretrievably lost its claim to the truth about nature, and had become the pursuit of necessary consequences of arbitrary axioms about meaningless things.”⁹¹

The extent of the relationship between mathematics and the “real” world continued to be a subject of intense debate. Some, such as Cantor saw the freedom from reality as that which distinguishes mathematics from other fields:

Mathematics is entirely free in its development and its concepts are restricted only by the necessity of being noncontradictory and coordinated to concepts previously introduced by precise definitions. ... The essence of mathematics lies in its freedom.⁹²

Others lamented “[t]he loss of truth and the seeming arbitrariness, the subjective nature of mathematical ideas and results”.⁹³ Felix Klein for example, himself a major figure in the development of non-Euclidean geometries, considered the investigation of such arbitrary structures as “the death of science”, and felt that “whoever has the privilege of freedom should also bear responsibility [of using it to investigate nature]”⁹⁴ Similarly, “Hilbert not only stressed that concrete problems are the lifeblood of mathematics, but took the trouble in 1900 to publish a list of twenty-three outstanding ones.”⁹⁵ Understandably, these differing viewpoints left an enduring division in the field, between so-called pure and applied mathematicians.⁹⁶

In the wake of all this arbitrariness, new attempts were being made to sure up the foundations of mathematics. Platonism remained popular during this period, since on a platonist account,

⁸⁹ Morris Kline, *Mathematical Thought from Ancient to Modern Times* (Oxford University Press, 1972) at 1016.

⁹⁰ Morris Kline, *Mathematical Thought from Ancient to Modern Times* (Oxford University Press, 1972) at 1034.

⁹¹ Morris Kline, *Mathematical Thought from Ancient to Modern Times* (Oxford University Press, 1972) at 1035.

⁹² Georg Cantor, “Über unendliche lineare Punktmannigfaltigkeiten” (1883) *Math. Ann* 21, cited in Morris Kline, *Mathematical Thought from Ancient to Modern Times* (Oxford University Press, 1972) at 1031. See also Jacobi, *Ges Werke*, 1, 454-55, cited in Kline at 1037: “It is true that Fourier is of the opinion that the principal object of mathematics is the public utility and the explanation of natural phenomena; but a scientist like him ought to know that the unique object of science is the honor of the human spirit and on this basis a question of [the theory of] numbers is worth as much as a question about the planetary system”.

⁹³ Morris Kline, *Mathematical Thought from Ancient to Modern Times* (Oxford University Press, 1972) at 1035.

⁹⁴ Morris Kline, *Mathematical Thought from Ancient to Modern Times* (Oxford University Press, 1972) at 1037.

⁹⁵ Morris Kline, *Mathematical Thought from Ancient to Modern Times* (Oxford University Press, 1972) at 1038.

⁹⁶ Morris Kline, *Mathematical Thought from Ancient to Modern Times* (Oxford University Press, 1972) at 1036. The division is complicated however by the way in which subjectively “pure” mathematics has underscored cutting-edge applications. For example, non-Euclidean geometries were key to Einstein’s development of his theory of relativity, since the theory requires that

space-time must be curved ... To describe such a four-dimensional curved space mathematically, the young physicist was looking for some kind of non-Euclidean geometry, and he found it in Riemann’s geometry, which allows for space to have a variable curvature. ... General relativity, therefore, can be said to be the final triumph of a mathematical idea which, in its infancy, was no more than an intellectual exercise.

non-Euclidean geometries exist, no less than Euclidean geometries. Correspondence with our experience of the physical world is not required. A number of additional schools emerged at this time, and now fall to be discussed.

Logicism The central tenet of logicism is that all of mathematics is merely a branch of logic. As such it is a form of rationalism. The logicist school was founded by Gottlob Frege, who hoped to show “that arithmetic is a branch of logic and need not borrow any ground of proof whatever from either experience or intuition.”⁹⁷ This “required that he show that the latter are derivable using only rules of inference, axioms, and definitions that are purely analytic principles of logic.”⁹⁸ Frege’s work on grounding mathematics in logic was almost complete when Bertrand Russell discovered a contradiction within its exposition of set theory,⁹⁹ an obstacle which Frege never surmounted.

Bertrand Russell and Alfred North Whitehead made their own subsequent attempt at grounding mathematics in logic. They used formalist methods¹⁰⁰ that all statements of number theory could be derived from logical axioms. In particular, Russell and Whitehead sought to avoid Russell’s paradox “by first creating a hierarchy of types, then assigning each mathematical ... entity to a type. Objects of a given type are built exclusively from objects of preceding types (those lower in the hierarchy), thus preventing loops.”¹⁰¹ However this theory of types gave rise to a great degree of additional complexity, which Russell and Whitehead sought to minimise through the introduction of an “axiom of reducibility”. This axiom was criticised as “a happy accident and not a logical necessity; it has been said that the axiom has no place in mathematics, and that what cannot be provided without it cannot be regarded as proved at all.”¹⁰² Such difficulties lend weight to Heyting’s observation that “[l]ogic is not the foundation of mathematics, on the contrary, it is conceptually a complicated and sophisticated part of mathematics.”¹⁰³ In any event, Russell and Whitehead’s system ultimately met with failure at the hand Kurt Gödel’s Incompleteness Theorem, which is discussed below.¹⁰⁴

⁹⁷ Gottlob Frege, *The Basic Laws of Arithmetic* (Montgomery Furth (trans), University of California, 1967) at §0, 29.

⁹⁸ Edward N Zalta, “Frege’s Logic, Theorem and Foundations for Arithmetic,” *Stanford Encyclopaedia of Philosophy*, 13 April 2007, <<http://plato.stanford.edu/entries/frege-logic/>> (27 March 2010) at 6.1.

⁹⁹ The paradox, known as Russell’s paradox, arose from the ability to define within Frege’s system a set of all the sets which were not members of themselves. Kline sources the contradiction not in Frege’s system, but in set theory itself. See Morris Kline, *Mathematical Thought from Ancient to Modern Times* (Oxford University Press, 1972) at 1183. Kline also sets out a popular form of the paradox as follows:

A village barber, boasting that he has no competition, advertises that of course he does not shave those people who shave themselves, but does shave all those who do not shave themselves. One day it occurs to him to ask whether he should shave himself. If he should shave himself, then by the first half of his assertion, he should not shave himself; but if he does not shave himself, then in accordance with his boast, he must shave himself. The barber is in a logical predicament.

¹⁰⁰ See the discussion of formalism below.

¹⁰¹ Wikipedia, “Type theory” <http://en.wikipedia.org/wiki/Type_theory> (30 November 2010).

¹⁰² Morris Kline, *Mathematical Thought from Ancient to Modern Times* (Oxford University Press, 1972) at 1197.

¹⁰³ Arend Heyting, “Intuitionistic views on the nature of mathematics” (1974) 27(1) *Synthese* 79 at 87. Heyting was a leading member of the intuitionist school, discussed below.

¹⁰⁴ Cf Bernard Linsky and Edward N Zalta, “What is Neologicism?” (2006) 12(1) *The Bulletin of Symbolic Logic* 60 at 61: “As we look back at logicism, we shall see that its failure is no longer such a clear-cut matter...”. The authors go on to argue a new breed of neologicism which “closely approximates the main goals of the original logicist programme.”

Chapter 5: The Nature of Mathematics

Formalism Formalists, led by David Hilbert, focused on expressing mathematics as formal logical systems and studying them without considering their meaning. To formalists, mathematics is “no more or less than mathematical language. It is simply a series of games...”¹⁰⁵ Mathematicians may read meanings into particular terms, but the terms themselves are without meaning. The truth value of mathematical theorems to a formalist, depends on their provability using the rules of the system. The success of any particular formal system depends on three things: consistency, completeness and decidability.¹⁰⁶

Some members of the logicist school attempted to use formalist methods to prove that mathematics is merely a branch of logic. In particular, Russell and Whitehead’s *Principia Mathematica*, claimed to base “all of mathematics on a logical system derivable from five primitive logical propositions whose truth is founded on basic intuition.”¹⁰⁷

A particularly painful thorn in the formalists’ side came in the form of a German mathematician, Kurt Gödel, who in proving Russell and Whitehead’s project flawed, came up with a mathematical proof called the incompleteness theorem, which can be summarised as follows:

Any sufficiently powerful formal system cannot be both complete and consistent.¹⁰⁸

Although Gödel’s proof only strictly extends to mathematical systems, Nagel and Newman suggest the theorem applies to “a very large class of deductive systems”.¹⁰⁹ Gödel dealt a fatal blow to the formalist program, at least as a foundation for mathematics.

However, the formalist notion of reducing the mental processes of ‘doing mathematics’ as a series of finite, mechanical steps continues to be important. In 1936, Alan Turing developed “a formal counterpart to the notion of a mental process,”¹¹⁰ the Turing machine, which became the theoretical model for the digital computer. So it seems that incompleteness and inconsistency aren’t holding the software industry back, as those who have done battle with a computer can surely attest.¹¹¹

Intuitionism Intuitionists, like LEJ Brouwer and Heyting, suggest that mathematics is “a creation of the human mind. Numbers, like fairy tale characters, are merely mental entities which would not exist if there were never any human minds to think about them.”¹¹² For intuitionists,

¹⁰⁵ WS Anglin, *Mathematics, a concise history and philosophy*, (Springer Verlag, 1994) at 218.

¹⁰⁶ “Consistency means that no contradictions will be found in the system, i.e., a theorem and its negation cannot both be true. Completeness implies that every theorem which is true in the system can be proved within the system. Decidability requires that a finite mechanical procedure exists that determines whether any given claim made by the system may be proved within the system.”: Brigham Narins (ed), *World of Mathematics*, Volume 1 (Gale Group, 2001) at 235.

¹⁰⁷ William P Berlinghoff, *Mathematics: The Art of Reason* (DC Heath & Company, 1968) at 203.

¹⁰⁸ The above simplification of the theorem is in this author’s words, but is based on a number of different formulations. See Douglas R Hofstadter, *Gödel, Escher Bach: An Eternal Golden Braid* (Basic Books, 1979); Brigham Narins (ed), *World of Mathematics*, Volume 1 (Gale Group, 2001) at 235.

¹⁰⁹ Ernest Nagel and James R Newman, *Gödel’s proof* (NYU Press, 2001) at 5.

¹¹⁰ Jean Lassègue, “Doing Justice to the Imitation Game; a Farewell to Formalism,” December 2003, <http://formes-symboliques.org/article.php?id_article=75> (10 September 2011).

¹¹¹ The relationship between Gödel’s theorem, and Turing’s model is the theoretical aspect of the isomorphism between mathematics and software development, which was touched on in Chapter 1 in Section 4 on page 28, and is explored further in Chapter 7 in Section 2 on page 277.

¹¹² WS Anglin, *Mathematics, a concise history and philosophy* (Springer Verlag, 1994) at 219.

the truth (or falsity) of mathematical propositions is not simply “discovered”, it must be experienced.¹¹³ As a foundation for mathematics, intuitionists start from the idea that “[t]he basic notions of mathematics are so extremely simple, even trivial, that doubts about their properties do not rise at all.”¹¹⁴ In particular, Heyting begins with an intuitive understanding of counting, which involves the isolation of one object after another. Since “[i]solating an object, focusing our attention on it, is a fundamental function of our mind,” he derives the notion that “[t]he entity conceived in the human mind is the starting point of all thinking, and in particular of mathematics.”¹¹⁵ Heyting then goes on to construct arithmetic, and then to deal with more complex mathematical systems such as the continuum, and set theory.¹¹⁶

One interesting aspect of the intuitionist position is that “[i]n its simplest form mathematics remains confined to one mind”.¹¹⁷ This means that mathematical formulae, which for the formalist are the ultimate expression of mathematics, are merely the language of mathematics. As with any language then, they are “not immune from misunderstanding”¹¹⁸ since “mental constructions cannot be rendered exactly by means of language”.¹¹⁹

The biggest criticism of the intuitionist position, a philosophy which at first seems quite plausible, is a pragmatic one. Intuitionists necessarily reject concepts like the infinite¹²⁰ and the law of the excluded middle.¹²¹ This makes it much harder to actually *do* practical mathematics.¹²²

¹¹³ “[O]n Brouwer’s view, there is no determinant of mathematical truth outside the activity of thinking, a proposition only becomes true when the subject has experienced its truth (by having carried out an appropriate mental construction); similarly, a proposition only becomes false when the subject has experienced its falsehood (by realizing that an appropriate mental construction is not possible). Hence Brouwer can claim that ‘there are no non-experienced truths’ ([LEJ Brouwer, *Collected Works*, Volume 1, Philosophy and Foundations of Mathematics (A Heyting (ed), North-Holland, 1975)], p.488).”: Mark van Atten, “Luitzen Egbertus Jan Brouwer” in Edward N Zalta (ed), *Stanford Encyclopaedia of Philosophy*, (Summer 2011 Edition) <<http://plato.stanford.edu/archives/sum2011/entries/brouwer/>> (10 September 2011).

¹¹⁴ Arend Heyting, “Intuitionistic Views of the Nature of Mathematics” (1974) 27 *Synthese* 79 at 79.

¹¹⁵ Arend Heyting, “Intuitionistic views on the nature of mathematics,” (1974) 27(1) *Synthese* 79, 80.

¹¹⁶ Arend Heyting, “Intuitionistic views on the nature of mathematics,” (1974) 27(1) *Synthese* 79, 81-88.

¹¹⁷ Arend Heyting, “Intuitionistic views on the nature of mathematics,” (1974) 27(1) *Synthese* 79 at 80.

¹¹⁸ Arend Heyting, “Intuitionistic views on the nature of mathematics,” (1974) 27(1) *Synthese* 79 at 88.

¹¹⁹ Arend Heyting, “Intuitionistic views on the nature of mathematics,” (1974) 27(1) *Synthese* 79 at 89. See also Morris Kline, *Mathematical Thought from Ancient to Modern Times* (Oxford University Press, 1972) at 1200-1201:

The world of mathematical intuition is opposed to the world of causal perceptions. In this causal world, not in mathematics, belongs language, which serves there for the understanding of common dealings. Words or verbal communications are used to communicate truths. Language serves to evoke copies of ideas in men’s minds by symbols and sounds. But thoughts can never be completely symbolized. ... Mathematical ideas are independent of the dress of language and are in fact much richer.

¹²⁰ The intuitionist position with respect to the infinite is perhaps more subtle than might be initially thought. Kline notes that “For Brouwer, as for all intuitionists, the infinite exists in the sense that one can always find a finite set larger than the given one. To discuss any other type of infinite, the intuitionists demand that one give a method of constructing or defining this infinite in a finite number of steps.”: Morris Kline, *Mathematical Thought from Ancient to Modern Times* (Oxford University Press, 1972) at 1202-1203.

¹²¹ This rule states that all things are either true or false. By excluding this rule, it is not possible for an intuitionist to prove the truth of a proposition by showing that it is not false.

¹²² Kline noted in 1972 that “[t]hey have succeeded in saving the calculus with its limit processes, but their construction is very complicated. They also reconstructed elementary portions of algebra and geometry. Unlike Kronecker, Weyl and Brouwer do allow some kinds of irrational numbers.”: Morris Kline, *Mathematical Thought from Ancient to Modern Times* (Oxford University Press, 1972) at 1203.

Chapter 5: The Nature of Mathematics

Summary As the 19th Century drew to a close, the various schools were in a standoff. No one theory had become dominant, and none appeared likely to become so. So as time went on, the impasse was literally worked around, with the foundations of mathematics being pushed to the side in order that mathematicians could get on with the *doing* of mathematics.

20th Century

In a sense, the development of mathematics over the 20th century continued the trends of the 19th century. Consistent with the move from Euclidean geometries to non-Euclidean geometries in physics (Newton to Einstein), “the emphasis [of mathematics] has shifted [from studying things on a small scale, in local co-ordinates] to try and understand ... global, large-scale behaviour.”¹²³ Other progressions involved the study of increased dimensions,¹²⁴ the shift from studying commutative to non-commutative systems,¹²⁵ from linear to non-linear systems.¹²⁶ These developments evidence the greater abstraction of mathematics away from the study of aspects of the physical world. The flip side of this abstraction was that these abstract models were much more complicated than anything which might form the subject of human experience of a three dimensional, linear world. A necessary consequence of this complexity was that the first half of the 20th century saw an increase in specialisation, as the foundational crisis in mathematics lead to its rigorous formalisation.¹²⁷ This in itself provides a challenge to the notion of proof, given that much of modern mathematics is beyond the expertise of all but the few mathematicians working in that area.¹²⁸

In contrast to the earlier half of the 20th century, the latter half “is much more ... the ‘era of unification’, where borders are crossed, techniques have been moved from one field into the

¹²³ Michael F Atiyah, “The Evolution of Mathematics in the 20th Century,” (2001) 108(7) *The American Mathematical Monthly* 654 at 654.

¹²⁴ In geometry, this meant a move beyond “things you could really see in space [to h]igher dimensions [which are] slightly fictitious, things that you could imagine mathematically”: Michael F Atiyah, “The Evolution of Mathematics in the 20th Century,” (2001) 108(7) *The American Mathematical Monthly* 654 at 656. In algebra, which was “always concerned with more variables” this meant moving “from finite dimensions to infinite dimensions, from linear space to Hilbert space, with an infinite number of variables”: Atiyah at 656.

¹²⁵ Such as matrices and quaternions. A matrix is “A rectangular array of symbols or mathematical expressions arranged in rows and columns, treated as a single entity”: “matrix” in *Oxford English Dictionary Online*, June 2011, <<http://oed.com/view/Entry/115057>> (10 September 2011). A quaternion, also known is a “four-dimensional hypercomplex number that consists of a real dimension and 3 imaginary ones (i, j, k) that are each a square root of -1. They are commonly used in vector mathematics and three-dimensional games.”: “Quaternion” in *Wiktionary*, 19 October 2010 <<http://en.wiktionary.org/wiki/quaternion>> (17 Nov 2010). All the laws of algebra apply to quaternions, except multiplication, which is non-commutative.

¹²⁶ Non-Euclidean geometries, discussed above, are an example of a non-linear system of geometry.

¹²⁷ The formalist approach to mathematics, discussed in further detail below, avoids the dependence on axioms as “‘self-evident truths’ ... in favour of an emphasis on such logical concepts as consistency and completeness”: “20th Century Mathematics” *The Story of Mathematics*, <<http://www.storyofmathematics.com/20th.html>> (18 November 2010). See also Michael F Atiyah, “The Evolution of Mathematics in the 20th Century,” (2001) 108(7) *The American Mathematical Monthly* 654 at 665.

¹²⁸ See Henk Barendregt and Freek Wiedijk, “The Challenge of Computer Mathematics” (2005) 363 *Philosophical Transactions: Mathematical, Physical and Engineering Sciences* 2351 at 2352: “During the course of history of mathematics [sic] proofs increased in complexity. In particular, in the 19th century, some proofs could no longer be followed easily by just any other capable mathematician: one had to be a specialist. This started what has been called the sociological validation of proofs [by peer review]. ... In the 20th century, this development went to an extreme.” The authors go on to note the proof of Fermat’s Last Theorem by Wiles was undertaken by a group of 12 specialist referees. The authors also note the proof of the Classification of the Finite Simple Groups in 1979 which “consisted of a collection of connected results written down in various places, totalling 10 000 page ... [and] also ‘well-known’ results [some of which turned out not to be valid]”: *ibid* at 2352. It was 2004 before the proof was finally settled.

other, and things have become hybridized to an enormous extent.”¹²⁹ Finally, it must be noted the extent to which mathematics has been, at least since the last quarter of the 20th century, influenced by developments in physics. Atiyah noted in 2001 that

[i]n the last quarter of the 20th Century, the one we have just been finishing, there has been a tremendous incursion of new ideas from physics into mathematics. ... The results predicted by the physicists have time and again been checked by the mathematicians and found to be fundamentally correct, even though it is quite hard to produce proofs and many of them have not yet been fully proved.¹³⁰

In the midst of all of this, it is germane to note one particularly important development in mathematics from early in the 20th century, the emergence of the theory of computability, and the development of the computer. In addition to having a use outside the field of mathematics, computers have come to be a useful tool in dealing with the increased complexity of mathematics, with some mathematical proofs about the development of computational mathematics, wherein computers are used to “prove” mathematical theorems which would otherwise involve too much computation.¹³¹ Computers have also provided a source of data for feeding mathematical intuitions in circumstances where exact proofs are difficult, or impossible.¹³²

In the wake of this hybridisation and increasing complexity of subject matter, it should come as no surprise that the corresponding developments in the philosophy of mathematics have become more subtle, and combine features of earlier schools of thought.

Quasi-empiricism Quine & Putnam’s *indispensability argument* claims to address the reliance of Platonism on faith alone by arguing “that the mathematics that is used in physical theories is confirmed along with those theories and that scientific realism entails mathematical realism”.¹³³ As such, this theory can be regarded as a mixture of empiricism and Platonism, hence the name quasi-empiricism. The Quine-Putnam argument can be summarised as follows:

Proposition 1: We ought to have ontological commitment to all and only the entities that are indispensable to our best scientific theories.

¹²⁹ Michael F Atiyah, “The Evolution of Mathematics in the 20th Century,” (2001) 108(7) *The American Mathematical Monthly* 654 at 665.

¹³⁰ Michael F Atiyah, “The Evolution of Mathematics in the 20th Century,” (2001) 108(7) *The American Mathematical Monthly* 654 at 663.

¹³¹ For example the Four-Colour problem and Kepler’s Conjecture. See the discussion in Henk Barendregt and Freek Wiedijk, “The Challenge of Computer Mathematics” (2005) 363 *Philosophical Transactions: Mathematical, Physical and Engineering Sciences* 2351 at 2352. For a discussion of the four-colour problem see Robert Tubbs, *What is a Number? Mathematical Concepts and Their Origins* (John Hopkins University Press, 2009) at 269-270. See James R Brown, *Philosophy of Mathematics: An Introduction to the World of Proofs and Pictures* (Routledge, 1999) at 154-158 for a discussion of the status of such a proof as a “new way of doing mathematics ... [which is] not a priori, [is] not certain, [is] not surveyable, and [is] not open to double-checking by other mathematicians”.

¹³² See James R Brown, *Philosophy of Mathematics: An Introduction to the World of Proofs and Pictures* (Routledge, 1999) at 158-171, wherein the author discusses conjectures and open problems in mathematics such as the qualities of “perfect” numbers, whether π is “normal”, and the Riemann hypothesis. Although the questions posed may seem more interesting than useful, the same may have been thought of non-Euclidean geometries before Einstein’s theory of relativity was developed.

¹³³ Susan Vineberg, “Confirmation and the Indispensability of Mathematics to Science” (1996) 63(3) *Philosophy of Science* S256 at S256.

Chapter 5: The Nature of Mathematics

Proposition 2: Mathematical entities are indispensable to our best scientific theories.

Conclusion: We ought to have ontological commitment to mathematical entities.¹³⁴

Of course this necessarily limits the justification of mathematics to “enough mathematics to serve the needs of science”.¹³⁵

The weak points of the indispensability arguments are generally held to lie within the first proposition.¹³⁶ Quine and Putnam found support for this proposition in a combination of naturalism and holism. Naturalism is the philosophical doctrine that “philosophy is neither prior to nor privileged over science”¹³⁷ based on a “deep respect for scientific methodology and an acknowledgment of the undeniable success of this methodology as a way of answering fundamental questions about all nature of things.”¹³⁸ However, naturalism “may or may not tell you whether to believe in all the entities of your best scientific theories”,¹³⁹ so holism¹⁴⁰ completes the picture by positing that “our statements about the external world face the tribunal of sense experience not individually but only as a corporate body”.¹⁴¹ Or put another way, “it is the same evidence that is appealed to in justifying belief in the mathematical components of the theory that is appealed to in justifying the empirical portion of the theory”.¹⁴²

Maddy¹⁴³ is critical of the indispensability thesis in that she sees a contradiction between the naturalist respect for the scientific method and holism. In particular, she notes that working scientists have a wide range of attitudes towards well-confirmed scientific theories, which “vary

¹³⁴ Adapted from Mark Colyvan, “Indispensability Arguments in the Philosophy of Mathematics” in Edward N Zalta (ed) *The Stanford Encyclopedia of Philosophy* (Spring 2011 Edition) <<http://plato.stanford.edu/archives/spr2011/entries/mathphil-indis/>> (28 June 2011).

¹³⁵ Mark Colyvan, “Indispensability Arguments in the Philosophy of Mathematics” in Edward N Zalta (ed) *The Stanford Encyclopedia of Philosophy* (Spring 2011 Edition) <<http://plato.stanford.edu/archives/spr2011/entries/mathphil-indis/>> (28 June 2011).

¹³⁶ However, for a critique of the second proposition, see Hartry H Field, *Science Without Numbers: A Defence of Nominalism*, (Blackwell, 1980). Fields’ program has itself been widely criticised. See for example Penelope Maddy, “Physicalistic Platonism” in AD Irvine (ed), *Physicalism in Mathematics* (Kluwer, 1990) at 259-289; David Malament, “Review of Field’s Science Without Numbers” (1982) 79(9) *Journal of Philosophy* 523; Michael D Resnik, “How Nominalist is Harry Field’s Nominalism” (1985) 47(2) *Philosophical Studies* 163; Stewart Shapiro, “Conservativeness and Incompleteness” (1983) 80(9) *Journal of Philosophy* 521.

¹³⁷ Mark Colyvan, “Indispensability Arguments in the Philosophy of Mathematics” in Edward N Zalta (ed) *The Stanford Encyclopedia of Philosophy* (Spring 2011 Edition) <<http://plato.stanford.edu/archives/spr2011/entries/mathphil-indis/>> (28 June 2011).

¹³⁸ Mark Colyvan, “Indispensability Arguments in the Philosophy of Mathematics” in Edward N Zalta (ed) *The Stanford Encyclopedia of Philosophy* (Spring 2011 Edition) <<http://plato.stanford.edu/archives/spr2011/entries/mathphil-indis/>> (28 June 2011).

¹³⁹ Mark Colyvan, “Indispensability Arguments in the Philosophy of Mathematics” in Edward N Zalta (ed) *The Stanford Encyclopedia of Philosophy* (Spring 2011 Edition) <<http://plato.stanford.edu/archives/spr2011/entries/mathphil-indis/>> (28 June 2011).

¹⁴⁰ Specifically confirmational holism, also called the Duhem-Quine thesis. See Mark Colyvan, “Indispensability Arguments in the Philosophy of Mathematics” in Edward N Zalta (ed) *The Stanford Encyclopedia of Philosophy* (Spring 2011 Edition) <<http://plato.stanford.edu/archives/spr2011/entries/mathphil-indis/>> (28 June 2011).

¹⁴¹ WV Quine, “Two Dogmas of Empiricism” (1951) 60 *The Philosophical Review* 20 at 38.

¹⁴² Mark Colyvan, “Indispensability Arguments in the Philosophy of Mathematics” in Edward N Zalta (ed) *The Stanford Encyclopedia of Philosophy* (Spring 2011 Edition) <<http://plato.stanford.edu/archives/spr2011/entries/mathphil-indis/>> (28 June 2011). See also Hilary Putnam, “What is Mathematical Truth”, in *Mathematics Matter and Method: Philosophical Papers*, Volume 1 (2nd ed, Cambridge University Press, 1979) at 74: “[M]athematics and physics are integrated in such a way that it is not possible to be a realist with respect to physical theory and a nominalist with respect to mathematical theory.”

¹⁴³ Penelope Maddy, “Indispensability and Practice” (1992) 89(6) *Journal of Philosophy* 275; Penelope Maddy, “Naturalism and Ontology” (1995) 3(3) *Philosophia Mathematica* 248. See also Penelope Maddy, *Naturalism in Mathematics* (Clarendon Press, 1997).

from belief, through tolerance, to outright rejection".¹⁴⁴ This attitude is incompatible with the all-or-nothing approach which confirmational holism requires. Similarly she notes that

Scientists seem willing to use strong mathematics whenever it is useful or convenient to do so, without regard to the addition of new *abstracta* to their ontologies, and indeed, even more surprisingly, without regard to the additional physical structure presupposed by that mathematics. On the one hand, they do not subject these mathematical and structural hypotheses to testing; on the other, they do not regard the empirical success of a theory using strong mathematics as confirming the mathematical or structural hypotheses involved.¹⁴⁵

On a similar note, Sober¹⁴⁶ attacks the empirical basis of confirmational holism by arguing that the mathematics used in science is not subject to the same testing as the empirical aspects of scientific theories. Further, the absence of alternatives to mathematics in supporting empirical theories suggests that mathematics is not confirmed by the empirical evidence. Despite these criticisms however, "the debate is very much alive, with many recent articles devoted to the topic."¹⁴⁷

Fictionalism Conceived by Field¹⁴⁸ as a response to Quine & Putnam's indispensability argument,¹⁴⁹ Field argued that whilst mathematics is undeniably useful, it is not essential to our understanding of the physical world.¹⁵⁰ To prove its dispensability, Field constructed a non-mathematical account of physics. Such an approach is interesting, because it sidesteps the issue of how mathematical propositions can be verified other than by empirical observation. Field's account, whilst recognised as a "major intellectual achievement",¹⁵¹ has been widely criticised.¹⁵²

¹⁴⁴ Penelope Maddy, "Indispensability and Practice" (1992) 89(6) *Journal of Philosophy* 275 at 280.

¹⁴⁵ Penelope Maddy, "Naturalism and Ontology" (1995) 3(3) *Philosophia Mathematica* 248 at 255.

¹⁴⁶ Elliot Sober, "Mathematics and Indispensability" (1993) 102(1) *Philosophical Review* 35.

¹⁴⁷ Mark Colyvan, "Indispensability Arguments in the Philosophy of Mathematics" in Edward N Zalta (ed) *The Stanford Encyclopedia of Philosophy* (Spring 2011 Edition)
<<http://plato.stanford.edu/archives/spr2011/entries/mathphil-indis/>> (28 June 2011).

¹⁴⁸ Hartry H Field, *Science Without Numbers: A Defence of Nominalism*, (Blackwell, 1980).

¹⁴⁹ Hartry H Field, *Science Without Numbers: A Defence of Nominalism*, (Blackwell, 1980) at 5.

¹⁵⁰ Hartry H Field, *Science Without Numbers: A Defence of Nominalism* (Blackwell, 1980) at 7-8.

¹⁵¹ Stewart Shapiro, *Thinking About Mathematics* (Oxford University Press, 2000) at 237.

¹⁵² For a summary of criticisms, see Mark Colyvan "Fictionalism in the philosophy of mathematics" in E.J. Craig (ed), *Routledge Encyclopedia of Philosophy Online edition*, (Taylor and Francis, 2011, forthcoming)
<<http://homepage.mac.com/mcolyvan/papers/fictionalism.pdf>> (11 March 2011). For a different criticism, noting a problem analogous to that which Gödel's incompleteness theorem caused for the formalist programme, see Stewart Shapiro, *Thinking About Mathematics* (Oxford University Press, 2000) at 235-236.

Chapter 5: The Nature of Mathematics

Social constructivism Social constructivists¹⁵³ see mathematics as merely “a social construction, a cultural product, fallible like any other branch of knowledge.”¹⁵⁴ On this view, truth in mathematics depends on “mathematical traditions, methods, problems, meanings and values into which mathematicians are enculturated – that work to conserve the historically defined discipline”.¹⁵⁵ In other words, mathematics is “constructed or created by – made real by – the activities of mathematicians.”¹⁵⁶ On such an understanding, the source of mathematical truth is “neither physical nor mental, it’s social. It’s part of culture, it’s part of history, it’s like law, like religion, like money, like all those very real things which are real only as part of collective human consciousness”.¹⁵⁷

Social constructivism is a kind of neo-Kantianism in which the objectivity of mathematics is sourced neither in the platonic realm, nor in the physical universe. Not surprisingly, this view is not warmly received by some mathematicians, who object to the notion that mathematical knowledge is merely relative, rather than a truly objective account. Others note that the statement that mathematics is a human activity is trivial, and adds nothing to the debate.¹⁵⁸ Azzouni puts forward a more nuanced objection, suggesting that an account based on social practices alone fails to explain the unique degree of conformity in mathematics:

I’m sympathetic to *many things* those who self-style themselves ‘mavericks’ have to say about how mathematics is a social practice. ... But many activities are similarly (epistemically) social: politicians ratify commonly-held beliefs and behaviour; so do religious cultists, bank tellers, empirical scientists and prisoners. ... It’s widely observed that, unlike other cases of conformity, and where social factors *really* are the source of that conformity, one finds in mathematical practice *nothing like* the

¹⁵³ The leading accounts of social constructivism in mathematics are Hersh and Ernest. See Reuben Hersh, *What is Mathematics, Really?* (Oxford University Press, 1997); and Paul Ernest, *Social Constructivism as a Philosophy of Mathematics* (State University of New York Press, 1998). For a summary of Hersh and Ernest’s views, and the difference between the two see Julian C Cole, “Mathematical Domains: Social Constructs?” in Bonnie Gold and Roger Simons (eds), *Proof and Other Dilemmas: Mathematics and Philosophy* (Mathematics Association of America, 2008). The social constructivist account is influenced by the philosophies of Wittgenstein, who (at least in his later accounts) saw mathematics as a language game, and Popper, who “saw mathematics (as well as science, art and other sociocultural institutions) as an evolutionary product of the intellectual efforts of humans who, by objectivizing their creations and trying to solve the often unintended and unexpected problems arising from those creations, produce new mathematical objections, problems and critical arguments”: Eduard Glas, “Mathematics as Objective Knowledge and as Human Practice” in Reuben Hersh (ed), *18 Unconventional Essays on the Nature of Mathematics* (Springer, 2006) at 289. For a summary of Wittgenstein’s philosophy of mathematics (and its misreading by some social constructivists), see Michael Peters “Wittgenstein, Education and the Philosophy of Mathematics” (2002) 3 *Theory & Science* <<http://theoryandscience.icaap.org/content/vol003.002/peters.html>> (6 December 2010).

¹⁵⁴ Paul Ernest, “Social Constructivism as a Philosophy of Mathematics: Radical Constructivism Rehabilitated?” (1990) <<http://people.exeter.ac.uk/PErnest/soccon.htm>> (10 September 2011), cited in Bonnie Gold and Roger Simons (eds), *Proof and Other Dilemmas: Mathematics and Philosophy* (Mathematics Association of America, 2008) at 39.

¹⁵⁵ Wikipedia, “Philosophy of Mathematics”, <http://en.wikipedia.org/wiki/Philosophy_of_mathematics> (20 April 2007).

¹⁵⁶ Julian C Cole, “Mathematical Domains: Social Constructs?” in Bonnie Gold and Roger Simons (eds), *Proof and Other Dilemmas: Mathematics and Philosophy* (Mathematics Association of America, 2008) at 111.

¹⁵⁷ John Brockman, “What Is Mathematics? A Talk With Reuben Hersh” *Edge* 5, 10 February 1997, <<http://edge.org/documents/archive/edge5.html>> (6 December 2010).

¹⁵⁸ See for example Hacking, who levels an attack against social constructivist accounts of the philosophy of science as a failing to precisely define the claims which such a philosophy is making. Ian Hacking, *The Social Construction of What?* (Harvard University Press, 1999).

variability found in cuisine, clothing, or metaphysical doctrine.¹⁵⁹

Structuralism Structuralists¹⁶⁰ such as Shapiro and Resnik see mathematics as “the science of structures”.¹⁶¹ The view might be called quasi-platonist, in that structuralists assert that “each unambiguous sentence of [mathematics] is true or false, independent of the language, mind, and social conventions of the mathematician.”¹⁶² Structuralists, unlike platonists however, deny that mathematics is composed of independent objects such as numbers, arguing that “[t]he objects of mathematics, that is, the entities which our mathematical constants and quantifiers denote, are structureless points or positions in structures. As positions in structures, they have no identity or features outside of a structure.”¹⁶³ The appeal of structuralism is that it corresponds with a greater emphasis on structures in modern mathematical practice.¹⁶⁴ However the summary just given does not address the ambiguity as to what is meant by the term structure, with the interpretations being “significantly different from each other, even conflicting in many ways.”¹⁶⁵ Further, each particular variant has its own philosophical problems, many of which are similar in nature to the issues facing the philosophies so far discussed.¹⁶⁶

Can these various views be reconciled?

As may have become apparent, these views are impossible to fully reconcile with each other,¹⁶⁷ and such a task is not attempted here. The more modest goal which is hoped to be achieved, and an important task for grounding the mathematics exception, is to set out a view of the non-patentability of mathematics which accommodates all of these various views. In other

¹⁵⁹ Jody Azzouni, “How and Why Mathematics is Unique as a Social Practice” in Reuben Hersh (ed) 18 *Unconventional Essays on the Nature of Mathematics* (Springer, 2006) at 201-202.

¹⁶⁰ See for example Michael D. Resnik, *Mathematics as a Science of Patterns* (1997, Clarendon Press); Stewart Shapiro, *Philosophy of Mathematics: Structure and Ontology* (1997, Oxford University Press). The emergence of structuralism is sometimes traced to Paul Benacerraf, “What Numbers Could Not Be” (1965) 74(1) *Philosophical Review* 47. A summary of the structuralist philosophy of mathematics can be found in Stewart Shapiro, in *Thinking About Mathematics* (Oxford University Press, 2000) ch 10; Erich H Reck and Michael P Price, “Structures and Structuralism in Contemporary Philosophy of Mathematics” (2000) 125 *Synthese* 341.

¹⁶¹ Stewart Shapiro, in *Thinking About Mathematics* (Oxford University Press, 2000) at 257.

¹⁶² Stewart Shapiro, in *Thinking About Mathematics* (Oxford University Press, 2000) at 257.

¹⁶³ Michael D Resnik, “Mathematics as a Science of Patterns: Ontology and Reference” (1981) 15 *Noûs* 529 at 530.

¹⁶⁴ Erich H Reck and Michael P Price, “Structures and Structuralism in Contemporary Philosophy of Mathematics” (2000) 125 *Synthese* 341 at 346.

¹⁶⁵ Erich H Reck and Michael P Price, “Structures and Structuralism in Contemporary Philosophy of Mathematics” (2000) 125 *Synthese* 341 at 374.

¹⁶⁶ For example, an obvious question arises as to what the nature of these structures is. These structures are variously defended on platonist, formalist and quasi-empiricist bases, introducing the same objections which those philosophies have sustained. For an overview of the main variants, and their individual difficulties, see Erich H Reck and Michael P Price, “Structures and Structuralism in Contemporary Philosophy of Mathematics” (2000) 125 *Synthese* 341.

¹⁶⁷ For example, one cannot be both a platonist, believing that mathematical objects exist in a place accessible only by the mind, and an empiricist, believing that mathematical truth is verified only by correspondence with the physical world. Nor can one be a logicist, believing mathematics is a branch of logic, and an intuitionist, believing that mathematics must be experienced to be verified. Some combinations are possible, for example, noting Russell’s combination of logicism and formalism. But a complete reconciliation remains impossible.

Chapter 5: The Nature of Mathematics

words, a post-modern,¹⁶⁸ holistic approach to reconciliation is taken.¹⁶⁹ Such an approach acknowledges that there is no one narrative which will provide the ultimate answer to the question of what mathematics is, but that each offers an insight into the nature of mathematics which captures something of the essence of mathematics, much like each of the 4 blind men captured something of the essence of the elephant.

Some support for a holistic view can be found in the attitudes of working mathematicians, since

most mathematicians work in their respective fields 'doing mathematics' and concern themselves very little with questions of philosophy. Each one has formulated an opinion about what constitutes mathematics that is sufficient to guide him in his research, and these opinions are often mixtures.¹⁷⁰

A good example of these mixtures can be seen in the philosophies of Gottlob Frege, who can be simultaneously identified with platonism,¹⁷¹ logicism, transcendental idealism,¹⁷² and whose work also greatly influenced formalism. Similarly, Heyting has noted:

There is no conflict between intuitionism and formalism when each keeps to its own subject, intuitionism to mental constructions, formalism to the construction of a formal system, motivated by its internal beauty or by its utility for science and industry.¹⁷³

Similarly, Avigad and Reck¹⁷⁴ note that despite Gödel's theorems, the formalist school still has a role to play in modern mathematics, in that it attempts to reconcile "general conceptual reasoning about abstractly characterized mathematical structures, on the one hand, and computationally explicit reasoning about symbolically represented objects, on the other."¹⁷⁵

¹⁶⁸ The author confesses to balking at describing the approach as post-modern, out of concern that post-modernism's distaste for meta-narratives, combined with the fact that it is itself a meta-narrative, is as paradoxical as a barber who shaves all those who do not shave themselves. Nonetheless, "postmodern is as good a name as any, especially since it's a bit of a joke on the ordinary meaning of modern. Obviously the Modern period was misnamed.": Larry Wall, "Perl, the first postmodern computer language" <<http://www.perl.com/pub/1999/03/pm.html>> (7 July 2011).

¹⁶⁹ In the present context, it is submitted that an inclusive course it to be preferred over a skeptical one. To quote Wittgenstein, "[w]e just *can't* investigate everything, and for that reason we are forced to rest content with assumption": Ludwig Wittgenstein, *On Certainty* (G.E.M Anscombe and G.H. von Wright, (trans), Wiley-Blackwell, 1975). The approach might be thought of as a variation of Quine's web theory: "The totality of our so-called knowledge or beliefs, from the most casual matters of geography and history to the profoundest laws of atomic physics or even of pure mathematics and logic, is a man-made fabric which impinges on experience only along the edges.": WV Quine, "Two Dogmas of Empiricism" (1951) 60(1) *The Philosophical Review* 20 at 39.

¹⁷⁰ William P Berlinghoff, *Mathematics: The Art of Reason* (1968) at 204.

¹⁷¹ "[T]he thought ... which we express in the Pythagorean theorem is timelessly true, true independently of whether anyone takes it to be true. It needs no bearer. It is not true for the first time when it is discovered, but is like a planet which, already before anyone has seen it, has been in interaction with other planets.": Gottlob Frege, "The Thought: A Logical Inquiry" (1956) 65 *Mind* 289 at 302.

¹⁷² "Frege defended Kant's philosophy of geometry, which he took to be consistent with logicism about arithmetic.": Lisa Shabel, "Kant's Philosophy of Mathematics" in Paul Guyer (ed), *The Cambridge Companion to Kant and Modern Philosophy* (2006) at 120, citing in support Gottlob Frege, *On the Foundations of Geometry and Formal Theories of Arithmetic* (Yale University Press, 1971).

¹⁷³ Arund Heyting, "Intuitionistic views on the nature of mathematics," (1974) 27(1) *Synthese* 79 at 89.

¹⁷⁴ Jeremy Avigad and Erich H Reck, "Clarifying the nature of the infinite: the development of metamathematics and proof theory," (Carnegie Mellon Technical Report No CMU-PHIL-120, 11 December 2001).

¹⁷⁵ Jeremy Avigad and EH Reck, "Clarifying the nature of the infinite: the development of metamathematics and proof theory," *Carnegie Mellon Technical Report*, CMU-PHIL-120, 11 December 2001, at 4.

It is submitted that the philosophy of mathematics, is like mathematics itself,

not a static body of revealed truth, but a complex of concepts in various stages of evolution, each of which is related to and affects all the others and contributes to their growth. By a process of consolidation and generalization, these concepts frequently merge and are submerged in more all-embracing concepts that emphasize what the mathematician frequently refers to as “the unity of mathematics.”¹⁷⁶

A holistic approach is beneficial in that it provides a way forward out of the less familiar territory of mathematical philosophy, back to the more comfortable realm of law. On an holistic approach, a theory of non-patentability should be considered to be a success, not because it accords with one particular view of mathematics, but because it can be substantially reconciled with *all* philosophies.

With this position in mind, it is time to consider the way in which mathematics has been considered by patent law, to determine whether a theory of non-patentability which can be reconciled with all theories of mathematics, already exists.

2 A Patent Lawyer's Account of Mathematics

With a broad appreciation of the nature of mathematics in mind, it is possible to critically assess the approach taken to mathematical subject matter in patent law. Although the European Patent Convention (“EPC”) expressly excepts mathematical methods ‘as such’ from patentability in Article 52, no such categorical exception to the patentability of all mathematical subject matter exists in either the US or Australia.¹⁷⁷ The approach which seems to be preferred is to address the patentability of mathematics by reference to other notions non-patentability. So although there may be agreement as to the inherent non-patentability of mathematics, it will be seen that patent law offers no cohesive explanation as to why it is the case.

It may be that the difficulty in finding an explanation as to the non-patentability of mathematics stems from the difficulties in finding a patent directed solely towards mathematical subject matter. Whether attributed to smart drafting by patent attorneys, or the diaphanous nature of mathematics, it is rare to find mathematical innovations claimed directly. More often, and as will be seen in the cases discussed below, the patentability of mathematics arises in the context of computer-implemented methods, data manipulation, signal processing and simulation/-modelling used in industrial design processes. In any event, if the heart of what is claimed lies in a mathematical innovation or advance, then it is necessary to consider the effect of any claimed exclusion of mathematics from the scope of patentability.

¹⁷⁶ Raymond M Wilder, “The Nature of Modern Mathematics,” in William E Lamon (ed), *Learning and the Nature of Mathematics* (Science Research Associates, 1972) at 47.

¹⁷⁷ Even within the EU, at the EPO mathematical subject matter could still be claimed so long as it involved the use of a pen and paper, since this would satisfy the technicality requirement. See T258/03 *Hitachi/Auction Method* [2004] EPOR 55 at [4.6], discussed in Chapter 3 at 116. With the majority in *Bilski* rejecting a categorical business method exception as ‘atextual’, the fate of a categorical mathematics exception would seem to be sealed in that jurisdiction as well. See Chapter 3 at 142 and following.

Chapter 5: The Nature of Mathematics

Some of the territory covered in this section will be familiar, given that it overlaps with the case law on computer-related inventions discussed already in Chapter 3. However, the focus here is slightly different, since the aim is to find an existing foundation on which to rest the non-patentability of mathematics which is consistent with the state of the law of patentable subject matter in each (or indeed any) of the jurisdictions so far surveyed.

Europe

It will be recalled from the discussion in Chapter 3 that under Article 52 of the EPC, mathematical methods are not patentable *as such*. It will also be recalled that the EPC exclusion also forms part of UK law, due to the inclusion of Article 52, the basis of the definition of patentable subject matter in s1(2)(c) *Patents Act 1977* (UK).

However, it will also be recalled that the words ‘as such’ in Article 52(2) have been the mechanism by which other exceptions, most notably the computer program exception, have been watered down by the European Patent Office¹⁷⁸ (“EPO”). In particular, the “any hardware” approach reduces the Article 52 exclusions to mere form requirements when claimed as implemented on a computers,¹⁷⁹ since any method “when implemented by a computer or other hardware apparatus ... is a man-made technical apparatus having a utilitarian purpose, and not a method at all, and for that reason avoids exclusion under article 52(2).”¹⁸⁰

European Patent Office

The position of the EPO on the patentability of mathematics is as follows:

These are a particular example of the principle that purely abstract or intellectual methods are not patentable. For example, a shortcut method of division would not be patentable but a calculating machine constructed to operate accordingly may well be patentable.¹⁸¹

The 2007 EPO case of *Circuit Simulation/Infineon Technologies*¹⁸² illustrates how mathematical methods, when run on a computer, are patentable following the “any hardware” approach. The invention in that case was for a “computer-implemented method for the numerical simulation of a circuit”.¹⁸³ It was claimed by the appellant inventor that the numerical simulation should be considered to meet the required standard of technical contribution on the basis that “technical considerations are required to solve problems in the engineering sciences, in particular electrical engineering ... to predict the performance of a circuit whose variables are

¹⁷⁸ See the discussion in Chapter 3 starting at 102.

¹⁷⁹ Or in any physical way. See the discussion of *Hitachi/Auction Method* in Chapter 3, 3 on page 115.

¹⁸⁰ Justine Pila, “Dispute over the Meaning of ‘Invention’ in Article 52(2) EPC – The Patentability of Computer-Implemented Inventions in Europe” (2005) 36 *International Review of Intellectual Property and Competition Law* 173 at 179.

¹⁸¹ European Patent Office, “Guidelines for Examination in the European Patent Office,” December 2007, <<http://www.epo.org/patents/law/legal-texts/guidelines.html>> (13 January 2008) at 2.3.3.

¹⁸² T1227/05 *Circuit Simulation/Infineon Technologies* [2007] OJ EPO 574 (“Infineon”).

¹⁸³ T1227/05 *Circuit Simulation/Infineon Technologies* [2007] OJ EPO 574 at 575 (Claim 1).

technical parameters. . .¹⁸⁴ The appellant further claimed that “the simulation . . . constitutes a technical process in itself”¹⁸⁵ since it created a model which “earlier technical literature holds to be difficult if not impossible”.¹⁸⁶ The appellant also claimed a technical contribution on the basis that the instruction in the claims “is addressed to the technical engineer, not the mathematician, and thus in itself constitutes technical teaching”.¹⁸⁷ Finally, the appellant claimed a technical contribution on the basis that this improved approach to modelling “requires shorter computing times and less storage space”¹⁸⁸ when implemented on a computer.¹⁸⁹

The Board reasoned that “[a]s the method according to independent claim 1 or 2 is computer-implemented, it uses technical means and by that very token has technical character”¹⁹⁰ In arriving at this conclusion, the Board was “persuaded that [the] simulation . . . constitute[d] an adequately defined technical purpose for a computer-implemented method, provided that the method is functionally limited to that technical purpose.”¹⁹¹ The technical purpose was to be found in the claimed simulation of “a circuit with input channels, noise input channels and output channels whose performance is described by differential equations”.¹⁹² As the veracity of the simulation could be verified by “the physical and mathematical derivation specified in the system”,¹⁹³ the board was “persuaded that the independent method claims are functionally limited”.¹⁹⁴

The Board also held that the invention was “neither a mathematical method as such nor a computer program as such, even if mathematical formulae and computer instructions are used to perform the simulation.”¹⁹⁵ This was so because although the simulation could be performed by a human, as a practical matter, the “simulation method cannot be performed by purely mental or mathematical means”.¹⁹⁶ Such simulations were “typical of modern engineering work”¹⁹⁷ and the increased efficiency of the claimed approach “enables a wide range of designs to be virtually tested and examined for suitability before the expensive circuit fabrication process starts.”¹⁹⁸ As such,

computer simulation methods for virtual trials are a practical and practice-oriented part of the electrical engineer’s toolkit. What makes them so important is that as

¹⁸⁴ T1227/05 *Circuit Simulation/Infineon Technologies* [2007] OJ EPO 574 at 577.

¹⁸⁵ T1227/05 *Circuit Simulation/Infineon Technologies* [2007] OJ EPO 574 at 578.

¹⁸⁶ T1227/05 *Circuit Simulation/Infineon Technologies* [2007] OJ EPO 574 at 578.

¹⁸⁷ T1227/05 *Circuit Simulation/Infineon Technologies* [2007] OJ EPO 574 at 578.

¹⁸⁸ T1227/05 *Circuit Simulation/Infineon Technologies* [2007] OJ EPO 574 at 578.

¹⁸⁹ Cf *Gale’s Patent Application* [1991] RPC 305, discussed below.

¹⁹⁰ T1227/05 *Circuit Simulation/Infineon Technologies* [2007] OJ EPO 574 at 581.

¹⁹¹ T1227/05 *Circuit Simulation/Infineon Technologies* [2007] OJ EPO 574 at 582.

¹⁹² T1227/05 *Circuit Simulation/Infineon Technologies* [2007] OJ EPO 574 at 582.

¹⁹³ T1227/05 *Circuit Simulation/Infineon Technologies* [2007] OJ EPO 574 at 582.

¹⁹⁴ T1227/05 *Circuit Simulation/Infineon Technologies* [2007] OJ EPO 574 at 582. See also 587, where the Board relevantly noted that the claims “both entail the specific modelling of an adequately defined class of technical systems (circuits) and define specific measures, not just mental constructs, for targeted implementation and application of the circuit model under the technically relevant conditions of 1/f noise.”

¹⁹⁵ T1227/05 *Circuit Simulation/Infineon Technologies* [2007] OJ EPO 574 at 582.

¹⁹⁶ T1227/05 *Circuit Simulation/Infineon Technologies* [2007] OJ EPO 574 at 583.

¹⁹⁷ T1227/05 *Circuit Simulation/Infineon Technologies* [2007] OJ EPO 574 at 583.

¹⁹⁸ T1227/05 *Circuit Simulation/Infineon Technologies* [2007] OJ EPO 574 at 583.

Chapter 5: The Nature of Mathematics

a rule there is no purely mathematical, theoretical or mental method that would provide complete and/or fast prediction of circuit performance ...¹⁹⁹

On this basis, the claimed method was held to have the requisite technical character. The decision is interesting on a number of fronts. Firstly, it illustrates the way that the any hardware approach makes brings nearly any conceivable abstract subject matter within the realm of the patentable. Secondly, the Board in its reasoning read into the “mathematical method” exclusion a requirement that the method be carried out by a human operator, thus excluding its relevance to computer-implemented methods. Finally, the patentability of mathematical innovations is clearly made dependent upon the audience to which such innovations are directed - if the innovation is likely to be of more interest to engineers than to mathematicians, then it is technical in character, and hence patentable.²⁰⁰

United Kingdom

As was seen in Chapter 3 in relation to software patenting, the UK position is less amenable to the patenting of mathematical methods, although the position in that jurisdiction is somewhat of a moving target.²⁰¹ According to the 4 step approach of *Aerotel*,²⁰² where the technical contribution is to the field of mathematics, such an invention will fall foul of the Article 52 exclusion. In other words, there will be a narrow focus on the nature of the advance when determining the application of the patentable subject matter exclusions. Such an approach, which attempts to divine the substance of the invention is inconsistent with the approach adopted at the EPO, where any mention in the claims of a physical manifestation, or technical purpose, will render the claimed subject matter patentable. In *Symbian*, this inconsistency was acknowledged, with the Court seemingly torn between maintaining consistency with the EPO on the one hand, but also looking to avoid “throwing the law into disarray”²⁰³ by adopting an interpretation which would mean that the exclusions have “lost all meaning”.²⁰⁴ There was some hope that some clear guidance might be garnered from the EPO by a reference of the issue to the Enlarged Board of Appeal. However, as noted in Chapter 3, the G3/08 referral avoided determining the issue, and the clash between the UK and EPO interpretations is far from resolved.

Gale’s Application *Gale’s Application*,²⁰⁵ discussed in Chapter 3,²⁰⁶ concerned an improved method of calculating the square root of a number, claimed as a ROM circuit designed to give effect to the method. On appeal to the Patents Court, Aldous J noted that

¹⁹⁹ T1227/05 *Circuit Simulation/Infineon Technologies* [2007] OJ EPO 574 at 583.

²⁰⁰ This reflects the claimed dichotomy between pure and applied mathematics, the difficulties of which was discussed above. It is also interesting to note that this result may be put another familiar way, namely that the mathematics in question produces a “useful result”.

²⁰¹ The position of the Court of Appeal in *Symbian*, discussed in Chapter 3 at 136, clearly indicates that the UK Courts maintain their current position until such time as “tolerably clear guidance” emerges from the EPO. Such guidance was no doubt expected from the G3/08 referral but, as discussed in Chapter 3 at 138, did not eventuate.

²⁰² *Aerotel Ltd v Telco Holdings Ltd (and others) and Macrossan’s Application* [2007] RPC 7, discussed in Chapter 3 at 131.

²⁰³ *Symbian Ltd v Comptroller-General of Patents* [2008] EWCA Civ 1066; [2009] RPC 1 at [46].

²⁰⁴ *Symbian Ltd v Comptroller-General of Patents* [2008] EWCA Civ 1066; [2009] RPC 1 at [46].

²⁰⁵ *Gale’s Patent Application* [1991] RPC 305.

²⁰⁶ See Section 3 on page 109.

the claim goes on to define [the ROM's circuitry] by the way it will be operated which in effect is a mathematical method of obtaining the square root of a number. No doubt the basis behind the claim can be said to be a mathematical method or a method for performing a mental act or even a program for a computer in that the ROM functions as a carrier or program which will be used in a computer."²⁰⁷

Despite this, his Honour held that because the claims were directed to a ROM, that this was different to a claim to the method itself, being:

a manufactured article having circuit connections which enables the program to be operated. A claim to a ROM with particular circuitry, albeit defined by functional steps, cannot to my mind be said to relate to the program or the functional steps as such.²⁰⁸

The Court of Appeal however rejected this interpretation, calling the claims to hardware a "confusing irrelevance",²⁰⁹ noting that "in substance, a claim to a series of instructions which incorporate Mr. Gale's improved method of calculating square roots."²¹⁰ The Court went on to reject Mr Gale's invention on a number of related grounds:

In the present case Mr Gale claims to have discovered an algorithm. Clearly that, as such, is not patentable. It is an intellectual discovery which, for good measure, falls squarely within one of the items, mathematical method, listed in section 1(2).²¹¹

The fact that it was a mathematical method was not dispositive of the issue. Nicholls LJ went on as follows:

[T]he nature of the discovery is such that it has a practical application, in that it enables instruction to be written for conventional computers in a way which will, so it is claimed, expedite one of the calculations frequently made with the aid of a computer. In my view the application of Mr. Gale's mathematical formulae for the purpose of writing computer instruction is sufficient to dispose of the contention that he is claiming a mathematical method as such.²¹²

However, the Court concluded that the invention could not escape the operation of the computer program exclusion, since the method neither embodied "a technical process which exists outside the computer",²¹³ nor did the instructions "solve a 'technical' problem lying within the computer".²¹⁴ Although the Court did not explain this conclusion further, the conclusion that

²⁰⁷ *Gale's Patent Application* [1991] RPC 305 at 316 per Aldous J.

²⁰⁸ *Gale's Patent Application* [1991] RPC 305 at 317 per Aldous J.

²⁰⁹ *Gale's Patent Application* [1991] RPC 305 at 326 per Nicholls LJ.

²¹⁰ *Gale's Patent Application* [1991] RPC 305 at 326 per Nicholls LJ.

²¹¹ *Gale's Patent Application* [1991] RPC 305 at 327 per Nicholls LJ.

²¹² *Gale's Patent Application* [1991] RPC 305 at 327 per Nicholls LJ.

²¹³ *Gale's Patent Application* [1991] RPC 305 at 327 per Nicholls LJ.

²¹⁴ *Gale's Patent Application* [1991] RPC 305 at 328 per Nicholls LJ.

Chapter 5: The Nature of Mathematics

the improved algorithm, although it “makes a more efficient use of a computer’s resources”²¹⁵ did not solve a “technical” problem is interesting. It is submitted that the lack of technicality refers the inquiry back again to the mathematics exception - an intellectual discovery or mathematical method such as that claimed cannot be technical.

Citibank v Comptroller of Patents The patentability of mathematical methods was directly considered in the *Citibank* case.²¹⁶ The claims under consideration concerned a method of managing risks associated with trade in financial derivatives by ensuring the integrity and validity of data used in such an evaluation.²¹⁷ The claimed method used a statistical analysis to compare the current data set with historical data to determine “the likelihood that changes to the set of input data are the result of one or more errors.”²¹⁸

Counsel for the Comptroller asserted that the claims were for a method, the substance of which was “two calculations and a statistical comparison and analysis”.²¹⁹ Counsel for Citibank “sought to draw a distinction between mathematical methods on the one hand and methods of calculation involving the application of mathematical methods on the other”,²²⁰ with the difference between the two said to reside in the fact that the former existed at a similarly “high level” to discoveries and scientific theories.²²¹

Mann J noted that no authority as to the meaning of the mathematical method exclusion was placed before the Court. After reviewing *Gale* and *Fujitsu*, his Honour noted that those cases did not support the sort of distinction sought to be made by Citibank, and held that the claims were a mathematical method “both in terms of the normal use of language and in terms of the likely policy underlying the Patents Act.”²²²

Mann J also flagged the relationship between mathematical methods and methods of performing mental acts, remarking that “the interrelationship between the concepts of a mental act and mathematical calculation might have assisted in arriving at a resolution of this matter, but am left with the appeal as presented to me.”²²³

The US

In contrast to the EU approach, it will be recalled that the codification of US patent law in §101 contains no express exceptions, instead using a positive definition of four types patentable subject matter, namely processes, machines manufactures and compositions of matter.

²¹⁵ *Gale’s Patent Application* [1991] RPC 305 at 327 per Nicholls LJ.

²¹⁶ *Citibank Na v The Comptroller General of Patents* [2006] EWHC 1676 (Ch) (09 June 2006).

²¹⁷ *Citibank Na v The Comptroller General of Patents* [2006] EWHC 1676 (Ch) (09 June 2006) at [2].

²¹⁸ *Citibank Na v The Comptroller General of Patents* [2006] EWHC 1676 (Ch) (09 June 2006) at [3].

²¹⁹ *Citibank Na v The Comptroller General of Patents* [2006] EWHC 1676 (Ch) (09 June 2006) at [19].

²²⁰ *Citibank Na v The Comptroller General of Patents* [2006] EWHC 1676 (Ch) (09 June 2006) at [20].

²²¹ *Citibank Na v The Comptroller General of Patents* [2006] EWHC 1676 (Ch) (09 June 2006) at [20]. What was meant by this “high level” was not further explained, but it seems safe to suggest that it is a reference to the abstract nature of the subject matter collected in this exclusion.

²²² *Citibank Na v The Comptroller General of Patents* [2006] EWHC 1676 (Ch) (09 June 2006) at [26].

²²³ *Citibank Na v The Comptroller General of Patents* [2006] EWHC 1676 (Ch) (09 June 2006) at [29]. In obiter his Honour dismissed a claim that for practical purposes the method would need to be run on a computer as not relevant. This was because the claims were clearly directed to mental acts, since they referred “to methods and techniques, not the physical means”: at [29].

Early jurisprudence

US jurisprudence on the patentability of mathematics demonstrates the slippery nature of mathematics. As determined above, there is considerable disagreement between mathematicians about what mathematics is. This uncertainty is also manifest in the case law. Some judgments attempt to equate mathematics with algorithms,²²⁴ in other instances mathematics is seen as “imperfect proxies for mathematical truths and other laws of nature.”²²⁵ For example, the 1939 US Supreme Court case of *Mackay Radio*²²⁶ focused on the communicative role mathematics often plays in science, and set in place the conceptual division between ‘discoveries’ and their ‘applications’. In this case, the US Supreme Court held that:

[w]hile a scientific truth, or the mathematical expression of it, is not patentable invention, a novel and useful structure created with the aid of knowledge of scientific truth may be.²²⁷

The applicant in that case sought to alter the mathematical formula describing his invention in order to cover a new configuration of antenna said to infringe his patent. The court refused the application on the grounds that where a scientific principle is expressed as a mathematical formula, altering the formula is not allowed where doing so would alter the nature of the law on which it was based.²²⁸ This court in this case clearly captured the notion of mathematics as a language to describe the attributes of nature. As such, it could be traced back to a bar on the patenting of scientific principles, as discussed in *O'Reilly v Morse*.²²⁹

Yuan

The case of *In re Yuan*,²³⁰ the issue of patentability and mathematics arose in a different context. In that case, the appellant had applied for a patent on a high speed airfoil with low drag characteristics.²³¹ However, the court determined that the appellant's contribution lay “in a mathematical procedure by which the aircraft designer can start with a pressure distribution curve of the required characteristics and convert it into a velocity distribution curve”.²³² As such, the invention was said to comprise of “purely mental steps dependent upon the mathematical formula which is recited in, and constitutes the heart of, the claims.”²³³ Since it had been “thoroughly established by decisions of various courts that purely mental steps do not

²²⁴ Most notably see *Gottschalk v Benson* 409 US 63 (1973), discussed in Chapter 3 at 3 on page 86. See the criticism therein by Newell, noting that it encourages an illusory distinction between the numerical and non-numerical.

²²⁵ John A Burtis, “Towards a Rational Jurisprudence of Computer-Related Patentability in Light of *In re Alappat*,” (1995) 79 *Minnesota Law Review* 1129 at 1157.

²²⁶ *Mackay Radio v RCA* 306 US 86 (1939).

²²⁷ *Mackay Radio v RCA* 306 US 86 (1939) at 94.

²²⁸ *Mackay Radio v RCA* 306 US 86 (1939) at 98.

²²⁹ *O'Reilly v Morse* 56 US 62 (1854).

²³⁰ 188 F.2d 377 (1951).

²³¹ *In re Yuan* 188 F.2d 377 (1951) at 378.

²³² *In re Yuan* 188 F.2d 377 (1951) at 379.

²³³ *In re Yuan* 188 F.2d 377 (1951) at 380.

Chapter 5: The Nature of Mathematics

form a process which falls within the scope of patentability as defined by statute”,²³⁴ the subject matter of the invention was not patentable.²³⁵

Benson and Diehr

The patentability of mathematics received its most detailed treatment in the US jurisdiction in the computer software cases. In the landmark case of *Gottschalk v Benson*,²³⁶ it was “held that the discovery of a novel and useful mathematical formula may not be patented”.²³⁷ The claimed invention was “a faster and more efficient mathematical procedure for transforming the normal ‘decimal’ type of numbers (base 10) into true ‘binary’ numbers (base 2) which are simpler to process within computers.”²³⁸ The inventors claims were “not limited to any particular art or technology, to any particular apparatus or machinery, or to any particular end use”²³⁹ although in argument before the court, the inventors’ attorney stated that the claim did not extend to its use by a human using a pen and paper.

In deciding the instant invention was non-patentable, the Court invoked earlier case law on the non-patentability of scientific principles,²⁴⁰ and also the non-patentability of ideas.²⁴¹ The Court concluded that “the ‘process’ claim [was] so abstract and sweeping as to cover both known and unknown uses”²⁴² and that as a result, the practical effect of allowing the patent would be that “the patent would wholly preempt the mathematical formula and, in practical effect, would be a patent of the algorithm itself”.²⁴³

This position was supported in the later Supreme Court case of *Diamond v Diehr* where the court said that “an algorithm, or mathematical formula, is like a law of nature, which cannot be the subject of a patent”.²⁴⁴

Alappat and State Street

Despite this seeming support for mathematics as non-patentable, inferior courts, and in particular the Federal Circuit, advanced the patentability of both software, and mathematics, by exploiting theoretical weaknesses in the *Benson* and *Diehr* judgements. As the Federal Circuit noted in *Alappat*:

²³⁴ 188 F.2d 377 (CCPA, 1951) at 380.

²³⁵ On the patentability of mental steps, see also *In re Abrams* 89 USPQ (BNA) 266 (1951). The mental steps doctrine was given short shrift in *In re Prater* 415 F.2d 1393 (1969), where the court held that the ability to perform the claimed process “wholly without human intervention”: at 1393, showed that it fell outside the doctrine. To the extent which the claims covered the performance of the process by mental steps, the court held the invention failed to comply with 35 USC 112 “which requires that the specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention”: at 1395.

²³⁶ 409 US 63 (1973).

²³⁷ *Parker v Flook* 437 US 584 (1978) at 585.

²³⁸ Oyez, “*Gottschalk v Benson*” <http://www.oyez.org/cases/1970-1979/1972/1972_71_485/> (26 Feb 2008).

²³⁹ *Gottschalk v Benson* 409 US 63 (1973) at 64.

²⁴⁰ See *Gottschalk v Benson* 409 US 63 (1973) at 67-68.

²⁴¹ “An idea, of itself, is not patentable.”: *Rubber-Tip Pencil Company v Howard* 87 US 498 (1874), cited in *Gottschalk v Benson* 409 US 63 (1973) at 67-68.

²⁴² *Gottschalk v Benson* 409 US 63 (1973) at 68.

²⁴³ *Gottschalk v Benson* 409 US 63 (1973) at 72.

²⁴⁴ *Diamond v Diehr* 450 US 175 (1981) at 186.

The Supreme Court has not been clear . . . as to whether such subject matter is excluded from the scope of 101 because it represents laws of nature, natural phenomena, or abstract ideas. See *Diehr*, 450 U.S. at 186 (viewed mathematical algorithm as a law of nature); *Gottschalk v. Benson*, 409 U.S. 63, 71-72 (1972) (treated mathematical algorithm as an “idea”). The Supreme Court also has not been clear as to exactly what kind of mathematical subject matter may not be patented. The Supreme Court has used, among others, the terms “mathematical algorithm,” “mathematical formula,” and “mathematical equation” to describe types of mathematical subject matter not entitled to patent protection standing alone. The Supreme Court has not set forth, however, any consistent or clear explanation of what it intended by such terms or how these terms are related, if at all.²⁴⁵

In *Alappat*, the court made it clear that it did not consider that mathematics constituted another category of non-patentable subject matter:

Mathematics is not a monster to be struck down or out of the patent system, but simply another resource whereby technological advance is achieved.²⁴⁶

As a result of this confusion, the law of patentable subject matter has moved on, to a point where many more recent decisions are hard to reconcile with a mathematics exception, such as *State Street*,²⁴⁷ where transformation of numerical data, put through a series of mathematical calculations by a computer constitutes patentable subject matter so long as it provides a “a useful, concrete, and tangible result”²⁴⁸ without pre-empting other uses of the algorithm. Much of this encroachment into the patentability of mathematics was achieved through the inclusion of a computer as a tangible component of the invention which executed the algorithm. However, in *AT&T v Excel* the need to claim a computer was removed, any application of mathematics which produced a “useful, concrete and tangible result” would be sufficient.²⁴⁹ That is, the inclusion of a computer in the claims, to run the algorithm, was no longer necessary. In the wake of *Bilski v Kappos* however, the continued applicability of this approach must be doubted. The court clearly noted that

nothing in today's opinion should be read as endorsing interpretations of §101 that the Court of Appeals for the Federal Circuit has used in the past. See, e.g., *State Street*, 149 F. 3d, at 1373; *AT&T Corp.*, 172 F. 3d, at 1357. It may be that the Court of Appeals thought it needed to make the machine-or-transformation test exclusive

²⁴⁵ *In re Alappat* 33 F.3d 1526 (1994) at 1543, footnote 19.

²⁴⁶ *In re Alappat* 33 F.3d 1526 (1994) at 1570.

²⁴⁷ *State Street Bank & Trust Co v Signature Financial Group Inc* 149 F.3d 1368 (1998).

²⁴⁸ “Today, we hold that the transformation of data, representing discrete dollar amounts, by a machine through a series of mathematical calculations into a final share price, constitutes a practical application of a mathematical algorithm, formula, or calculation, because it produces “a useful, concrete and tangible result” — a final share price momentarily fixed for recording and reporting purposes and even accepted and relied upon by regulatory authorities and in subsequent trades.”: *State Street Bank & Trust Co v Signature Financial Group Inc* 149 F.3d 1368 (1998) at 1373.

²⁴⁹ “Whether stated implicitly or explicitly, we consider the scope of § 101 to be the same regardless of the form — machine or process — in which a particular claim is drafted.”: *AT&T Corp v Excel Communications Inc* 172 F.3d 1352 (1999) at 1357.

Chapter 5: The Nature of Mathematics

precisely because its case law had not adequately identified less extreme means of restricting business method patents, including (but not limited to) application of our opinions in *Benson*, *Flook*, and *Diehr*.²⁵⁰

Pre-Bilski practice

The current USPTO Manual of Patent Examining Procedure²⁵¹ (“MPEP”) does not adopt the useful result approach, and suggests instead that claimed inventions involving mathematics will be non-statutory (non-patentable) where they:

- consist solely of mathematical operations without some claimed practical application (i.e., executing a “mathematical algorithm”); or
- simply manipulate abstract ideas, e.g., a bid (*Schrader*, 22 F.3d at 293-94, 30 USPQ2d at 1458-59) or a bubble hierarchy (*Warmerdam*, 33 F.3d at 1360, 31 USPQ2d at 1759), without some claimed practical application.²⁵²

Following the MPEP, it would seem that the only limit to the patentability of a mathematical innovation is one’s ability to imagine a practical application for it. Mathematicians have long noted the impossibility of drawing distinctions between pure and applied mathematics, in that almost every area of mathematics, no matter how “pure” it may have been considered at one time or another, has been found to have a practical application. The MPEP also goes on to explain the relationship between laws of nature and mathematical algorithms in the following terms:

Certain mathematical algorithms have been held to be nonstatutory because they represent a mathematical definition of a law of nature or a natural phenomenon. For example, a mathematical algorithm representing the formula $E = mc^2$ is a “law of nature” - it defines a “fundamental scientific truth” (i.e., the relationship between energy and mass). To comprehend how the law of nature relates to any object, one invariably has to perform certain steps (e.g., multiplying a number representing the mass of an object by the square of a number representing the speed of light). In such a case, a claimed process which consists solely of the steps that one must follow to solve the mathematical representation of $E = mc^2$ is indistinguishable from the law

²⁵⁰ *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3223 per Kennedy J. Stevens J went further, noting that relying on the useful effect approach would be a “grave mistake”: at 3232. Similarly, Breyer J noted that the rejection of the machine-or-transformation test as the sole test for patentability “by no means indicates that anything which produces a ‘useful, concrete, and tangible result,’ is patentable.”: at 3259 per Breyer J (citations omitted). His Honour noted his earlier statement in *Laboratory Corp. of America Holdings v Metabolite Laboratories Inc* 548 US 124 (2006) at 136, that “this Court has never made such a statement and, if taken literally, the statement would cover instances where this Court has held the contrary.”

²⁵¹ United States Patent and Trademark Office, “2106.02 Mathematical Algorithms,” Manual of Patent Examining Procedure, Eighth Edition, August 2001, Revision 5, August 2006
<http://www.uspto.gov/web/offices/pac/mpep/documents/2100_2106_02.htm> (8 September 2011).

²⁵² United States Patent and Trademark Office, “2106.02 Mathematical Algorithms,” Manual of Patent Examining Procedure, Eighth Edition, August 2001, Revision 5, August 2006
<http://www.uspto.gov/web/offices/pac/mpep/documents/2100_2106_02.htm> (8 September 2011).

of nature and would “pre-empt” the law of nature. A patent cannot be granted on such a process.²⁵³

Klemens noted that despite continuing theoretical support for the exclusion of mathematics “standing alone”, the US judiciary had, prior to *Bilski* at least, enabled the patenting of mathematics. He gives examples of the following 4 patents:

- Method and system for solving linear systems (U.S. Patent No. 6078938).
- Cosine algorithm for relatively small angles (No. 6434582).
- Method of efficient gradient computation (No. 5886908).
- Methods and systems for computing singular value decompositions of matrices and low rank approximations of matrices (No. 6807536).²⁵⁴

Bilski v Kappos

It will be recalled from Chapter 3 that the invention claimed in *Bilski* was “a method of hedging risk in the field of commodities trading”²⁵⁵. *Bilski v Kappos* must therefore be taken to reinforce the non-patentability of mathematics, as the subject matter of the claims was clearly a mathematical algorithm, including data gathering and processing, without any claims to a physical instantiation designed to carry the method out. The invention was characterised by the majority as being directed to a non-patentable abstract idea.²⁵⁶ Further, the existence of a “useful result” will no longer be of relevance, as the *State Street* approach was clearly rejected.²⁵⁷

Previous USPTO practice must now be read subject to the interim guidelines issued in the wake of *Bilski v Kappos*.²⁵⁸ These guidelines specifically recite mathematical concepts as an example of a general concept amounting to a non-patentable abstract idea.²⁵⁹ Relatedly, the Guidelines also note the non-patentability of mental activity, teaching concepts, and human behaviour (which is said to include “following rules or instructions”).²⁶⁰

The Guidelines do leave open the possibility of patenting machine-implemented mathematical algorithms, although this is likely to depend on both the particularity of the machine cited,

²⁵³ United States Patent and Trademark Office, “2106.02 Mathematical Algorithms,” Manual of Patent Examining Procedure, Eighth Edition, August 2001, Revision 5, August 2006
<http://www.uspto.gov/web/offices/pac/mpep/documents/2100_2106_02.htm> (8 September 2011).

²⁵⁴ Ben Klemens, “Software Patents Don’t Compute,” *IEEE Spectrum Online*, July 2005,
<<http://spectrum.ieee.org/careers/careerstemplate.jsp?ArticleId=i070305>> (4 January 2008).

²⁵⁵ *In re Bernard L. Bilski and Rand A. Warsaw* (2008) 545 F.3d 943 at 949.

²⁵⁶ *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3230 per Kennedy J.

²⁵⁷ *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3231 per Kennedy J.

²⁵⁸ United States Patent and Trademark Office, “Interim Guidelines for Determining Subject Matter Eligibility Claims in View of *Bilski v Kappos*” (2010) 75(143) *Federal Register* 43,922.

²⁵⁹ United States Patent and Trademark Office, “Interim Guidelines for Determining Subject Matter Eligibility Claims in View of *Bilski v Kappos*” (2010) 75(143) *Federal Register* 43,922 at 43,926.

²⁶⁰ United States Patent and Trademark Office, “Interim Guidelines for Determining Subject Matter Eligibility Claims in View of *Bilski v Kappos*” (2010) 75(143) *Federal Register* 43,922 at 43,926.

Chapter 5: The Nature of Mathematics

and the extent to which the machine imposes meaningful limitations upon the claimed algorithm.²⁶¹ Similarly, the transformation limb of the “machine-or-transformation” test does allow the possibility of patenting computer-implemented mathematics, to the extent that the implementation would transform a general purpose computer into a specific-purpose computer, along the lines discussed in Chapter 4.²⁶²

Australia

Mathematics is not included in the express exceptions in *Patents Act 1990* (Cth) s18, nor was it included in the *Statute of Monopolies 1623* (UK) on which the Australian definition of patentability relies.²⁶³ Nevertheless, “mathematical algorithms ... have traditionally been regarded as not per se patentable, because they do not exhibit the requirements of a manner of manufacture”.²⁶⁴

Australian patentable subject matter law has been consistently moving away from express categorisations, towards a general test for patentability (*NRDC*).²⁶⁵ The most likely determinant of the patentability of a mathematical innovation depends on whether it is both:

- an artificially created state of affairs, and
- of utility in the field of economic endeavour.

As most discussions of mathematics tends to occur in the context of computer software patenting, the former is more or less assumed, and the focus of the second limb becomes a question of whether the mathematical algorithm in question produces a ‘useful effect’. The most recent statement of this requirement comes from *Grant v Commissioner of Patents*,²⁶⁶ where the Full Court of the Federal Court held:

It has long been accepted that ... a mathematical algorithm ... *without effect* [is] not patentable. ... It is necessary that there be some “useful product”, some physical phenomenon or effect resulting from the working of a method for it to be properly the subject of letters patent.²⁶⁷

The rejection of the patentability of a legal structure, and the divination of a requirement of physicality which accords with the *Bilski* position noted above seem to suggest that the non-patentability of mathematics is likely to continue in this jurisdiction. However, this is not a

²⁶¹ United States Patent and Trademark Office, “Interim Guidelines for Determining Subject Matter Eligibility Claims in View of *Bilski v Kappos*” (2010) 75(143) *Federal Register* 43,922 at 43,925.

²⁶² United States Patent and Trademark Office, “Interim Guidelines for Determining Subject Matter Eligibility Claims in View of *Bilski v Kappos*” (2010) 75(143) *Federal Register* 43,922 a 43,925, specifically factor B(3).

²⁶³ It will be recalled from Chapter 2 at 38, that s18 (1)(a) requires that for an invention to be patentable it must be “a manner of manufacture within the meaning of section 6 of the Statute of Monopolies”.

²⁶⁴ Australian Patent Office, “2.9.2.5 Discoveries, Ideas, Scientific Theories, Schemes and Plans” in *Manual of Practice and Procedure* <http://www.ipaustralia.gov.au/pdfs/patentsmanual/WebHelp/Patent_Examiners_Manual.htm> (10 September 2011).

²⁶⁵ William van Caenegem, *Intellectual Property* (2nd ed, LexisNexis Butterworths, 2006) at 126-127.

²⁶⁶ (2006) 154 FCR 62.

²⁶⁷ *Grant v Commissioner of Patents* (2006) 154 FCR 62 at 73 (emphasis added).

certainty, as the position in Australia mirrors that of the US as espoused in *State Street*. If nothing else, *IBM v Commissioner of Patents*,²⁶⁸ held by the Full Federal Court in *Grant* to be consistent with their physical result requirement, is in fact a patent on purely mathematical subject matter. In that claimed invention, a set of numerical control points were put through a couple of known mathematical algorithms to produce a table of numbers which could be put into memory to produce a curve on a screen. This was held to be a manner of manufacture as it produced a commercially useful effect in the field of computer graphics.

The inputs to the claimed algorithm were as follows:

A set of control points which define the curve and which are input for each dimension and a number of intervals of the curve to be computed...²⁶⁹

The aspect of the algorithm most focused on however was the output, or effect:

[I]t is not suggested there is anything new about the mathematics of the invention. What is new is the application of the selected mathematical methods to computers, and in particular, to the production of the desired curve by computer. This is said to involve steps which are foreign to the normal use of computers and, for that reason, to be inventive. The production of an improved curve image is a commercially useful effect in computer graphics.²⁷⁰

The problem with accepting this analysis is that the actual display of the curve on the screen of a computer is taken care of by an entirely independent set of components, namely the video card (display adapter). To understand why this is the case, a basic understanding of how things are drawn on the screen is necessary. Each of the dots on a computer screen, or pixels, is represented in the memory of the computer as a single value. The screen is represented in a contiguous series of memory addresses which map out a 'virtual screen' table corresponding to the dots on the table. A simple 9x9 monochrome (single colour) screen can be represented by a 10x10 table of pixel values, with 0 being black and 1 being white. A blank screen would be represented as shown in Table 5.1.

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Table 5.1: Blank screen

²⁶⁸ (1991) 33 FCR 218.

²⁶⁹ *IBM v Commissioner of Patents* (1991) 33 FCR 218 at 4 (Claim 1).

²⁷⁰ *IBM v Commissioner of Patents* (1991) 33 FCR 218 at 225-226.

Chapter 5: The Nature of Mathematics

By updating the values stored in the table, a simple circle could be represented by changing some of the pixel values in memory as shown in Table 5.2 (the zeros are represented by blank cells).

			1	1	1	1			
		1					1		
	1							1	
	1							1	
	1							1	
	1							1	
		1					1		
			1	1	1	1			

Table 5.2: A circle on the screen

At the time of the case, a typical screen size was 1024x768 pixels, with a range of 65536 colours available for each pixel (called 16-bit colour depth). To accommodate this, a much larger table is required, but the concepts remain the same. Colours are handled by storing an integer between 0 and 65535 in the table. A 16-bit pixel value actually contains the binary values of 3 colour channels – red, green and blue, as shown in Table 5.3.

Red (5 bits)					Green (6 bits)						Blue (5 bits)				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Table 5.3: 16-bit colour representation

After the display memory is set, the display adapter reads the virtual screen data. At the time of IBM's case, these digital pixel values would have to be then passed through a digital-to-analogue converter built in to the video card:

The modern display comes from a long lineage of cathode ray tubes (CRTs). A CRT display uses an electron gun to blast three different materials on the inside of the tube that emit red, green and blue light when excited. These early devices were analog by nature and to convert from digital to analog a device called a digital to analog converter (DAC) made its way into graphics outputs.²⁷¹

A 'virtual screen' table such as that produced by the IBM algorithm could through similar means be sent to a printer instead of a screen. The operation of the display adapter and digital analog converter are entirely independent of the claimed algorithm here, and it is wrong to treat them as a part of the claimed invention. If this element is not considered to be part of the patented invention, then we are left with a process which has as its inputs, a series of numbers representing control points of a curve. These inputs are put through a number of mathematical

²⁷¹ Don Woligroski, "The Basic Parts of a Graphics Card" *Tom's Hardware*, 24 July 2006 <<http://www.tomshardware.com/reviews/graphics-beginners,1288-2.html>> (15 October 2010).

transformations, to arrive at a table of integer numbers. Clearly such an invention is in essence mathematical in nature.

Why is mathematics non-patentable?

Armed with the requisite knowledge of the nature of mathematics, derived from the history and philosophies of mathematics discussed above, it might be hoped that it would be a simple matter to find a suitable (or at least consistent) basis upon which to rest the non-patentability of mathematics. There are numerous bases upon which it might be sought to set such a claim. The case law considering mathematics frames the discussion in various ways, including:

- discoveries, or intellectual discoveries²⁷²
- scientific theories, or the expression thereof²⁷³
- an intellectual discovery²⁷⁴
- abstract information, abstract ideas²⁷⁵
- mental steps²⁷⁶

It might be thought that one (or various) of these descriptions, arising as they do from the exposition of the concept of patentable subject matter over time in the jurisdictions considered, would be possible candidates upon which to base the non-patentability of mathematics. However, most of these explanations are problematic for one reason or another.

Some of the claims meet immediate objections based on the nature of mathematics discussed in Section 1. For example, the classification of mathematics truths as concerning scientific truths or discoveries, whilst consistent with “realist” philosophies such as empiricism and platonism cannot be reconciled with constructivist accounts, which if accepted would suggest that mathematics is created, and hence may be patentable where of economic use. The latter position raises an issue of whether mathematics can properly be considered to be something “under the sun that is made by man”²⁷⁷ or an “artificially created state of affairs”.²⁷⁸ This suggests that the notion of discovery might be synonymous with the notion of abstract idea, or an absence of technical character.

²⁷² The notion of mathematics as an non-patentable discovery extends back at least as far as to 1884, when in *Young v Rosenthal* (1884) 1 RPC 29 at 31, Grove J referred to mathematics as an “abstract discovery”. Mathematics has been couched in similar terms since that time. This correspondence between discoveries and mathematics is enshrined in Article 52 of the EPC, where mathematical methods are listed together with discoveries and scientific theories. See also *Citibank Na v The Comptroller General of Patents* [2006] EWHC 1676 (Ch) (09 June 2006) at [20] where Mann J notes that all these subjects occur at a similarly “high level”. The latter term, “intellectual discovery” was applied to the inventor’s algorithm for determining the square root of a number in *Gale’s Patent Application* [1991] RPC 305 at 327 per Nicholls LJ.

²⁷³ *Mackay Radio v RCA* 306 US 86 (1939) at 94.

²⁷⁴ *Gale’s Patent Application* [1991] RPC 305 at 327 per Nicholls LJ.

²⁷⁵ For example, *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3230 per Kennedy J; *Citibank Na v The Comptroller General of Patents* [2006] EWHC 1676 (Ch) (09 June 2006).

²⁷⁶ *In re Yuan* 188 F.2d 377 (CCPA, 1951).

²⁷⁷ *Diamond v Chakrabarty* 447 US 303 (1980) at 309.

²⁷⁸ *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 277.

Chapter 5: The Nature of Mathematics

Similarly, attempting to distinguish between discoveries, theories, intellectual information or abstract ideas and their application is problematic because of the interrelationship of so-called ‘pure’ and ‘applied’ mathematics:

What is considered applied mathematics today may, by a curious reversal of process, become pure mathematics tomorrow. And at any given moment of time, there is no clear distinction between what is pure and what is applied. I have even noticed how two groups of mathematicians, each of which considers itself applied, have each denied the propriety of the designation “applied” to the the fellow group. And ... even the mathematician who insists he is a pure mathematician is in reality an applied mathematician in that his interests are applications to the conceptual world of mathematics.²⁷⁹

Similarly, Lobachevsky said that “[t]here is no branch of mathematics, however abstract, which may not some day be applied to phenomena of the real world.”²⁸⁰ The non-Euclidean geometries such as that which Lobachevsky developed are a case in point.²⁸¹ Similarly, as noted by Atiyah above, mathematics at the turn of the 21st century is greatly informed by, and therefore closely related to modern developments in physics.²⁸²

Even the most impressive of candidates have problems. The interrelationship between mathematical methods and mental steps was referred to favourably by the Court in the UK case of *Citibank*,²⁸³ and was dispositive of the issue in the US case of *Yuan*.²⁸⁴ As noted in Chapter 4, the doctrine showed great promise in the software context in that it brought into issue the relationship between the human mind and the computer.²⁸⁵ Given that mathematics is largely performed without a computer, and noting the prominence of thought processes inherent in the notions of abstraction, deduction and proof discussed above, it might be thought that such a doctrine would be particularly useful in addressing the non-patentability of mathematics. But the abandonment of the doctrine in the US in favour of a focus on “mathematical algorithms” means that any patentability put on such a basis stands on shaky ground.²⁸⁶

²⁷⁹ Raymond M Wilder, “The Nature of Modern Mathematics,” in William E Lamon (ed), *Learning and the Nature of Mathematics* (Science Research Associates, 1972) at 46.

²⁸⁰ Cited in George E Martin, *The foundations of geometry and the non-Euclidean plane* (Springer, 1998) at 225.

²⁸¹ “No more impressive warning can be given to those who would confine knowledge and research to what is apparently useful, than the reflection that conic sections were studied for eighteen hundred years merely as an abstract science, without regard to any utility other than to satisfy the craving for knowledge on the part of mathematicians, and that then at the end of this long period of abstract study, they were found to be the necessary key with which to attain the knowledge of the most important laws of nature.”: Alfred North Whitehead, *Introduction to Mathematics* (Williams & Northgate, 1911) at 136-137.

²⁸² Claims to a clear distinction between pure and applied science in general have been questioned by Stokes, who notes an additional class of use-inspired basic research. See Donald E Stokes, *Pasteur’s Quadrant - Basic Science and Technological Innovation* (Brookings Institution Press, 1997).

²⁸³ *Citibank Na v The Comptroller General of Patents* [2006] EWHC 1676 (Ch) (09 June 2006). Mann J noted at [27] that “closeness of those concepts was reflected by Laddie J in *Fujitsu Limited’s Application* [1996] RPC 511, 532” and that to his Honour’s mind, “the interrelationship between the concepts of a mental act and mathematical calculation might have assisted in arriving at a resolution of this matter”: at [30].

²⁸⁴ *In re Yuan* (1951) 38 CCPA 967.

²⁸⁵ See Chapter for at 154 onwards.

²⁸⁶ At the time of completing this thesis, the mental steps doctrine seems to be getting a new lease of life in *CyberSource Corporation v Retail Decisions Inc* Appeal No 2009-1358 (Fed Cir, 2011). The limitations of the mental steps approach are discussed in Chapter 8.

Similarly, the now dominant notion in the current US approach, that of the non-patentable abstract idea, which seems to acknowledge the fundamental role which abstraction plays in mathematics, is not without its problems. The notion of abstraction, and the corresponding machine-or-transformation 'clue' call into question the importance of manifesting invention in a physical form. As Stevens J noted in *Bilski*, where the claims were dismissed as non-patentable abstract ideas, the majority failed to provide "a satisfying account of what constitutes a non-patentable abstract idea".²⁸⁷ This author would add to his Honour's criticism, noted that there exists a failure to adequately explain *why such abstract ideas should remain non-patentable*. It is further submitted that this is the reason why a distinction between discovery and invention "is not precise enough to be other than misleading".²⁸⁸ The lack of precision in these concepts arises from a lack of proper explanation of the policy which informs them.

In all the jurisdictions just surveyed, it seems to be accepted that, in theory if not in practice, mathematics is not patentable. And in its favour, the mathematics exception has lasted nearly 400 years. But the practical reality is at odds with the theory. Given the trend towards expansionism,²⁸⁹ and the demise of the business method exception both in Australia and in the US, the lack of a well understood mathematics exception is a cause for concern, especially given the usefulness of mathematical advances to modern product markets such as:

- bioinformatics
- computing
- cryptography
- finance
- robotics
- image processing
- nanotechnology

This concern is particularly acute when one considers the fate of the business method exception. Although not expressly specified, such an exception was held to exist for a long time.²⁹⁰ However, in the US and Australia, this category of patentable subject matter was first undermined, and then removed. In *State Street*, as noted in Chapter 3, the business method exception was described as "ill-conceived"²⁹¹ and summarily dismantled. Although *State Street* has been similarly wiped out by *Bilski v Kappos*, the majority's textual approach leaves no room for a

²⁸⁷ *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3236.

²⁸⁸ *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 264.

²⁸⁹ It will be recalled from Chapter 3 that a broadening conception of the boundaries of patent law has been a characteristic of patent law over the last century, with the tide only now seeming to retreat.

²⁹⁰ "Although it is difficult to derive a precise understanding of what sorts of methods were patentable under English law, there is no basis in the text of the *Statute of Monopolies*, nor in the pre-1790 English precedent, to infer that business methods could qualify." *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3240 per Stevens J. In relation to the subsequent development of US law, Stevens J, after a detailed analysis noted that "the historical clues converge on one conclusion: A business method is not a 'process'": at 3250.

²⁹¹ *State Street Bank & Trust Co v Signature Financial Group Inc* 149 F.3d 1368 (1998) at 1375.

Chapter 5: The Nature of Mathematics

resurrection of this categorical exception.²⁹² There is similarly nothing in the text of the the US Patents Act or Constitution which bars the patenting of mathematics.

In Australia, Heerey J in *Catuity*,²⁹³ following the US approach from *State Street*, held that there was no reason to maintain an exception for business methods, preferring instead that applications for patents in this area be judged upon the standard criteria for patentability in line with *NRDC*. If similar logic were to be applied to mathematics, then a mathematical advancement could be considered patentable so long as it is described by reference to a “concrete, tangible, physical or observable effect”²⁹⁴ Given how IBM was demonstrated to be directed to mathematical subject matter above, this suggests that this new test is in reality only a form requirement.

It is submitted that this mismatch between the theory and reality arises from a fundamental misunderstanding of the reason why mathematics stands outside the patent regime. If the reasons for excluding mathematics are not well understood, then it is to be expected that support for it will be gradually eroded. But the very persistence of the exception to this point suggests that it reflects important factors at work in the patent regime. Through attempting to highlight the reasons why mathematics is considered non-patentable, this thesis will attempt to highlight some of these inner workings of the patent system, and further inform debate about the dimensions of patent law. This thesis will attempt to reconcile the differences between a mathematical and patent law understanding of mathematics, by looking not to the question of what mathematics *is*, but to what mathematics *requires* – the optimal conditions for its advancement.

3 Conclusion

This chapter has compared a mathematical understanding of the nature of mathematics, derived from the historical and philosophical accounts of mathematics, with a legal understanding of the nature of mathematics as set out in case law in the three jurisdictions considered throughout this thesis.

Whilst it has been made clear that the issue of the nature of mathematics is not one which is likely to be resolved, it is through understanding the various accounts together that the limitations of present explanations of mathematics’ non-patentability can be seen. It has been demonstrated that no explanation thus far put forward is acceptably wide that it can accommodate all present understandings of mathematics.

As such, the next chapter builds upon this understanding of the nature of mathematics and the limitations of previous accounts of its non-patentability, by looking at the issue in a different fashion – not at what mathematics is, but at what it requires for mathematical innovation to prosper. This is surely the right question, since it is towards the promotion of that innovation to which patent law is said to be directed. The next chapter will use these historical and

²⁹² “The Court is unaware of any argument that the “‘ordinary, contemporary, common meaning,” *Diehr* ... at 182, of ‘method’ excludes business methods.”: *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3221 per Kennedy J.

²⁹³ *Welcome Real-Time SA v Catuity Inc and Ors* (2001) 113 FCR 110.

²⁹⁴ *Grant v Commissioner of Patents* (2006) 154 FCR 62 at 70.

Conclusion

philosophical accounts of the nature of mathematics just considered, to build a set of common features found therein. From these commonalities, it will be argued why mathematical innovation depends on freedom to continue to advance. From that understanding, an explanation of its non-patentability will be advanced.

Why Mathematics Is Not Patentable

“There was a blithe certainty that came from first comprehending the full Einstein field equations, arabesques of Greek letters clinging to the page, a gossamer web. They seemed insubstantial when you first say them, a string of squiggles. Yet to follow the delicate tensors as they contracted, as the superscripts paired with subscripts, collapsing mathematically into concrete classical entities – potential; mass; forces vectoring in a curved geometry – that was a sublime experience. The iron fist of the real: inside the velvet glove of airy mathematics.”¹

1 Introduction

It has been seen that legal accounts of the nature of mathematics considered thus far do not sit well with the various philosophies of mathematics. Some explanations fall short of the mark, and some, whilst showing promise and capturing some aspect of the nature of mathematics, give at best a partial account. This chapter proposes to take a different approach. Whereas the historical, philosophical and legal accounts of mathematics considered have centred on what mathematics *is*, it will be seen that the better focus, is on what mathematics *requires* for its further progress. It will be seen that by looking at the requirements of mathematicians, that is, by focusing on the nature of mathematical development, or mathematical innovation, it is possible to avoid difficult questions as to the ontological status of mathematical objects. In that way it is possible to avoid the difficulty of providing a definitive unifying account of what mathematics is, but nonetheless to answer the question of whether mathematics is a proper subject for patent protection.

It is asserted that the basis of mathematical progress is freedom, namely the ability of mathematicians to create new mathematics free of constraints. Without that freedom, the advancement of mathematics would become impossible. More specifically, it is submitted that the mathematics exception should be understood as a recognition of the utmost importance of

¹ Gregory Benford (1941) in Lloyd Albert Johnson, *A Toolbox for Humanity: More than 9000 Years of Thought* (Trafford Publishing, 2004) at 89.

Chapter 6: Why Mathematics Is Not Patentable

freedom of thought and expression to the continued development of mathematics. It will be shown that this freedom is evident from the cognitive and expressive aspects which lie at its core.

At this point it should be acknowledged that it is not the need for freedom which of itself which distinguishes mathematics from other fields. To an extent, freedom is a prerequisite of progress in all fields of human endeavour. However, it is the *extent* of the freedom required which, it is submitted, makes mathematics distinctive from other fields which fall within the ambit of patentable subject matter. This chapter therefore explores the role which freedom plays in mathematics, leading back towards a proper explanation of its non-patentability consistent with traditional notions of patentability. What is asserted is that the level of freedom required by mathematics in order for it to advance is greater than the level of freedom which features in the patent paradigm.

In exploring the role of freedom in mathematical advancement, three specific aspects of that freedom are noted. First, the importance of freedom of thought underscores the fundamentally abstract rather than physical nature of mathematics. Second, the need for freedom of expression is borne out by the notion of mathematics as a language. As a language, mathematics is expressive rather than purposive. Although mathematics may be put to work in useful processes, its essence lies in its ability to express our understanding of those processes. In other words, it is the symbolic nature of mathematics that lies at its core. Third, the need for expression also draws out the aesthetic rather than rational nature of mathematics. The end product of mathematical activity, mathematical proofs, presents as sequences of logical deductions from accepted axioms. But this logical, rational nature belies the creative process by which it is forged.

These three aspects, the abstract, expressive and aesthetic, pave the road back to patent law. The traditional division between the fine and useful arts, via a foray into the philosophy of technology, provides the basis upon which an explanation of mathematics' non-patentability is constructed. In simple terms, it will be shown that mathematics is not to be patentable because it has the characteristics of the fine arts, rather than the useful arts. It will also be shown that this explanation of non-patentability can be reconciled with philosophies of mathematics discussed in the Chapter 5. This explanation of mathematics' non-patentability, and the analytical tools derived in coming up with that explanation will be applied to the issue of the patentability of software in the next chapter.

2 Freedom in patent law

The contemporary understanding of the role of patent law is as an incentive to innovation.² The mechanism by which patent law seeks to incentivise innovation is through the promise of monopoly rights in exchange for disclosure of technological advances.³ The monopoly right awarded to the patent grantee, being in the nature of a proprietary right to exclude, is based

² "The essence of the patent system is to encourage entrepreneurs to develop and commercialise new technology": Commonwealth of Australia, "Patents Bill 1990: Second Reading" Senate, 29 May 1990.

³ See *Liardet v Johnson* (1778) 1 Carp Pat Cas 35 (NP), discussed in Chapter 2 at 3.

upon a paradigm of control. A consequence of this grant of control, is a loss of freedom of competitors, who are excluded from the area of invention defined by the boundaries of the patent. Thus patent law can be characterised as an attempt to balance “conflicting public interests of a proper reward for investors on the one hand without an unreasonable fetter on the freedom of third parties on the other.”⁴ To say this is to draw attention to the social contract theory of patent law, put forward in *Liardet v Johnson*,⁵ which suggests loss of those freedoms is the cost borne by the public in exchange for the award of monopolies to encourage advancement. The emergence of a social contract theory of patent law at the same time as broader notions of a social contract,⁶ and the importance of the historical investigation of manner of manufacture, suggests that reference to freedom as understood now and then is appropriate. However, a structural bias exists in patent law, which has a tendency to focus attention on the former interest at the expense of the latter:

We know that [law] built around the self-interest of existing and aspirant monopolists will protect a variety of private goods, namely those of the firms and interests at the table. We know also that it will fail to protect certain kinds of interests – most notably those of large numbers of unorganised individuals with substantial collective, but low individual, stakes in the matters being discussed.⁷

Using the paradigm of freedom developed in human rights jurisprudence to define and draw attention to the interests of these unrepresented many, is therefore an important way of addressing the imbalance. It should however be noted that the reference to human rights jurisprudence will not be used to expound a reconception of intellectual property based entirely in human rights, as as a paradoxical competition between a right to property versus other human rights.⁸ Rather, an understanding of the impact of monopoly on those other than the inventor/creator must be understood as one effect of the award of patent rights. What is sought to be relied upon is the explanatory power of these notions to aid in the provision of a satisfactory account of a class of excluded subject matter in the context of the patentable subject matter inquiry.

⁴ Thomas Terrell and Simon Thornley, *Terrell on Patents*, (15th ed, Sweet & Maxwell, 2000) at 13.

⁵ (1778) 1 Carp Pat Cas 35 (NP), discussed in Chapter 2, on page 45.

⁶ Social contract theory was first put forward by Thomas Hobbes, in *Leviathan* (1651). According to Hobbes, it is through an act of consent to a social contract individual wherein each “lay down this right to all things; and be contented with so much liberty against other men, as he would allow other men against himselfe” and thereby created a civil society. John Locke’s version of the theory was set out in set out in his *Second Treatise of Government* (1689). Locke’s conception involved a similar grant by individuals of their liberty, but whereas Hobbes’ grant to the sovereign was irrevocable and absolute, Locke’s individuals granted only “the power necessary to the ends for which they unite into society.” (Locke, *Second Treatise*, VIII, §99), which had an ongoing limiting effect on State power.

⁷ James Boyle, “Enclosing the Genome: What the Squabbles over Genetic Patents Could Teach Us” in F Scott Kieff, *Perspectives on the Human Genome Project* (Academic Press, 2003) 97 at 117. This was exactly the way in which the TRIPS treaty came into being. See Peter Drahos with John Braithwaite, *Information Feudalism: Who Owns the Knowledge Economy?* (Earthscan Publications, 2002).

⁸ In particular it is accepted that “there is little reason to think that the human rights concerns associated with creative labor must be furthered by recognizing a right to full control over the information that creative labour produces”: Rochelle C Dreyfuss, “Patents and Human Rights: Where is the Paradox” in William Grosheide (ed), *Intellectual Property and Human Rights: A Paradox* (Edward Elgar, 2010) 72 at 73.

Chapter 6: Why Mathematics Is Not Patentable

3 What is meant by freedom?

Freedom is a concept which has attracted the attention of countless poets, philosophers and statesmen. But the concern of this thesis is not this broad conception of freedom as “the quality or state of being free”,⁹ but freedom as “a political or civil right”.¹⁰ In particular, it is the jurisprudence of human rights, namely freedom of thought, and the closely related right of freedom of expression which require further investigation.

Freedom of Thought

A useful starting point is the definition given to it in the *Universal Declaration of Human Rights*,¹¹ Article 18:

Everyone has the right to freedom of thought, conscience and religion; this right includes freedom to change his religion or belief, and freedom, either alone or in community with others and in public or private, to manifest his religion or belief in teaching, practice, worship and observance.¹²

Philosophical justifications of freedom of thought find their source in the 18th century, during the Enlightenment, which was “characterised by a rejection of knowledge derived through tradition and authority, including religious authority. Instead, the fundamental source of knowledge became reason – that is, the capacity of human beings to know truth through independent and critical thought.”¹³ J.S. Mill was a strong adherent of freedom of thought, which he thought should include “absolute freedom of opinion and sentiment on all subjects, practical or speculative, scientific, moral or theological”¹⁴ In many ways, freedom of thought is the font from which all other freedoms flow:

Freedom of thought ... is the matrix, the indispensable condition, of nearly every other form of freedom. With rare aberrations a pervasive recognition of this truth can be traced in our history, political and legal.¹⁵

As a consequence, it is unsurprising that a close relationship between freedom of thought and freedom of expression exists.¹⁶ In the introduction to *A History of Freedom of Thought*,¹⁷ Bury describes the relationship as follows:

⁹ “freedom” in *Merriam-Webster’s Dictionary of Law* (Merriam-Webster, 1996).

¹⁰ “freedom” in *Merriam-Webster’s Dictionary of Law* (Merriam-Webster, 1996).

¹¹ *Universal Declaration of Human Rights*, GA Res 217A (III), UN GAOR, 3rd sess, 183rd plen mtg, UN Doc A/810 (10 December 1948) (“the UDHR”).

¹² The language of the article also makes clear the close link between freedom of thought and freedom of religion. Freedom of religion is relevant only as an example of a manifestation of a thoughts on a particular topic. The right is similarly defined in Article 19, *International Covenant on Civil and Political Rights*, opened for signature 16 December 1966, 999 UNTS 171 (entered into force 23 March 1976).

¹³ Lawrence McNamara, “Chapter 1: Free Speech,” in Des Butler and Sharon Rodrick, *Australian Media Law* (2nd ed, Lawbook Co, 2004) at 5. This is of course the same sentiment which underpinned the rationalist school generally. On rationalism in mathematics see Chapter 5 at 198 and below.

¹⁴ John Stuart Mill, *On Liberty* (J.W. Parker and Son, 1859) at 11.

¹⁵ *Palko v State of Connecticut* 302 US 319 (1937) at 327.

¹⁶ Similarly, note also the inclusion of a freedom to *manifest* thought in the UDHR definition above.

¹⁷ John B Bury, *A History of Freedom of Thought* (Williams and Norgate, 1914).

It is a common saying that thought is free. A man can never be hindered from thinking whatever he chooses so long as he conceals what he thinks. The working of his mind is limited only by the bounds of his experience and the power of his imagination. But this natural liberty of private thinking is of little value. It is unsatisfactory and even painful to the thinker himself, if he is not permitted to communicate his thoughts to others, and it is obviously of no value to his neighbours. Moreover it is extremely difficult to hide thoughts that have any power over the mind. If a man's thinking leads him to call in question ideas and customs which regulate the behaviour of those about him, to reject beliefs which they hold, to see better ways of life than those they follow, it is almost impossible for him, if he is convinced of the truth of his own reasoning, not to betray by silence, chance words, or general attitude that he is different from them and does not share their opinions. Some have preferred, like Socrates, some would prefer today, to face death rather than conceal their thoughts. Thus freedom of thought, in any valuable sense, includes freedom of speech.¹⁸

The interconnectedness of thought and language, and indeed the proposition that language determines thought,¹⁹ was similarly expounded by J.S. Mill²⁰ and popularised by George Orwell in his novel *1984*.²¹ The link between thought and speech was also explored by the psychologist Vygotsky, by studying the development of children. He concluded as follows:

It would be wrong ... to regard thought and speech as two unrelated processes, either parallel or crossing at certain points and mechanically influencing each other. The absence of a primary bond does not mean that a connection between them can be formed only in a mechanical way ... The meaning of a word represents such a close amalgam of thought and language that it is hard to tell whether it is a phenomenon of speech or a phenomenon of thought. A word without meaning is an empty sound; meaning, therefore, is a criterion of "word," its indispensable component. It would seem then that it may be regarded as a phenomenon of speech. But from the point of view of psychology, the meaning of every word is a generalization or a concept. And since generalizations and concepts are undeniably acts of thought, we may regard meaning as a phenomenon of thinking.²²

¹⁸ John B Bury, *A History of Freedom of Thought* (Williams and Norgate, 1914) at 7-8.

¹⁹ Such a notion is known as linguistic determinism. See for example Benjamin Whorf, "Science and Linguistics" (1940) 42 *Technology Review* 229. Support for such a notion can also be found in the analytic philosophy of Wittgenstein: "The limits of my language mean the limits of my world.": Ludwig Wittgenstein, *Tractatus Logico-Philosophicus* (Cosimo, 2010), at 88 (Proposition 5.6).

²⁰ John Stuart Mill, *On Liberty*, (J.W.Parker and Son, 1859) at 11-12: "The liberty of expressing and publishing opinions may seem to fall under a different principle, since it belongs to that part of the conduct which concerns other people, but, being almost of as much importance as the liberty of thought itself and resting in great part on the same reasons, is practically inseparable from it."

²¹ In 1984, the government invent a new language called "Newspeak", a language designed "not only to provide a medium of expression for the world-view and mental habits proper to the devotees of Ingsoc, but to make all other modes of thought impossible.": George Orwell, "Appendix: The Principles of Newspeak", *Nineteen Eighty Four*, <<http://www.netcharles.com/orwell/books/1984-Appendix.htm>> (16 April 2008).

²² Lev S Vygotsky and Alex Kozulin *Thought and Language* (2nd ed, MIT Press, 1986) at 211-212.

Chapter 6: Why Mathematics Is Not Patentable

Because of the difficulties in actually directly restricting the thoughts of another, and also because of the lack of consensus on exactly what thought is,²³ it is difficult to make any authoritative pronouncements as to the nature of this freedom. Given the relationship between thought and expression just discussed, attention is often focused on the indirect protection of thought through the protection of expression.²⁴ However, as will be seen below, both aspects draw attention to different parts of the practice of mathematics, and as such will be separately considered.

Freedom of Expression

Freedom of expression, sometimes referred to as freedom of speech, is also defined in the UDHR, in Article 19:

Everyone has the right to freedom of opinion and expression; this right includes freedom to hold opinions without interference and to seek, receive and impart information and ideas through any media and regardless of frontiers.²⁵

Freedom of expression can be justified by its relationship to a set of “basic purposes, aims, or goals thought to be pursued by our constitutional protection of freedom of speech itself.”²⁶ Whilst there may be many ways of categorising the various values which free speech is said to protect, the primary justifications are based on autonomy, truth and democracy.²⁷ Each of these will now be considered in turn.

²³ Debates as to what thought is are “as old as human beings and as fascinating as life itself.” Anil K Rajvanshi, *Nature of Human Thought* (NARI, 2004) at 4.

²⁴ This is also the approach of the European Court of Human Rights, who “prefer to examine an applicant’s complaints solely under Article 10 [freedom of speech] if possible.”: John Wadham et al, *Blackstone’s Guide to the Human Rights Act 1998* (4th ed, 2007). See *Paturel v France*, App No 54968/00, 22 December 2005.

²⁵ The link noted earlier between freedom of expression and freedom of thought is immediately apparent, by virtue of the inclusion of freedom of opinion. Article 19, *International Covenant on Civil and Political Rights*, opened for signature 16 December 1966, 999 UNTS 171 (entered into force 23 March 1976) is drawn in similar terms.

²⁶ R George Wright, “Why Free Speech Cases Are As Hard (And As Easy) As They Are” (2001) 68 *Tenn. L. Rev.* 335 at 337.

²⁷ This classification was adopted after reading Lawrence McNamara, “Chapter 1: Free Speech,” in Des Butler and Sharon Rodrick, *Australian Media Law* (2nd ed, Lawbook Co, 2004), although it generally accords with the three justifications put forward in other sources. See for example R George Wright “Why Free Speech Cases Are as Hard (And as Easy) as They Are” (2001) 68 *Tennessee Law Review* 335; Keith Werhan, *Freedom of Speech: A Reference Guide to the United States Constitution* (Greenwood Publishing Group 2004) at 28. According to Baker, noted US free speech scholar Thomas Emerson put forward for 4 justifications for free speech, namely, “(1) individual self-fulfillment, 2) advancement of knowledge and discovery of truth, 3) participation in decision-making by all members of society ([including] the right to participate in the building of the whole society), and 4) achievement of a more adaptable and hence stable community” but it is submitted that the third and fourth categories are adequately considered together under the heading of democracy. See C. Edwin Baker, *Human Liberty and Freedom of Speech* (Oxford University Press, 1989) at 47. Susan Williams starts with these three traditional justifications, but notes that the democratic justification is “ultimately derivative of the truth theory, or the autonomy theory, or both.” See Susan Hoffman Williams *Truth, Autonomy, and Speech: Feminist Theory and the First Amendment* (NYU Press, 2004) at 27-31.

Truth

J.S. Mill's *On Liberty*²⁸ is the leading exposition of the notion of freedom of expression. Within a utilitarian framework,²⁹ he argued for an almost unbridled freedom of speech. This was based on "a sort of conceptual Darwinism: the conviction that in a 'free market of ideas', the best will come to the fore and survive."³⁰ Specifically, Mill offered three arguments for the importance of the free expression of ideas:

First, he suggested that the idea suppressed as false may in fact be true, since to contend otherwise is to assume the infallibility of the individuals who adhere to the dominant opinion. Second, he argued that the suppressed opinion might be at least partially true, since one view rarely contains all of the truth in a given area. Finally, Mill suggested that even if the suppressed idea were completely false, its suppression would tend to result in the true idea's becoming a sterile and unchallenged dogma that would lack the vital force necessary for a living truth.³¹

This 'truth' goes beyond a scientific correctness, to include "political or ethical truths".³² The value of the truth justification lies in the expression itself, wherein any "epistemic advance" has value.³³ As such, free speech puts value on "truths, half-truths, gross errors and vividly and emptily held truths of many sorts; in politics, culture and entertainment, as well as science; and in preparatory as well as public or final forms of expression."³⁴

Critics of the truth theory have pointed out that it is by no means certain that truth will prevail over falsity,³⁵ that this assumption rests on a "false confidence in the rationality of human beings"³⁶ and the claim that there is such thing as objective truth.³⁷ Yet despite this, the truth theory continues to be an influential factor in free speech jurisprudence.

²⁸ John Stuart Mill, *On Liberty* (J.W.Parker and Son, 1859).

²⁹ Mill argued that the best utilitarian outcome resulted from a minimal regulation of individual liberty: see John Stuart Mill, *On Liberty*, (J.W.Parker and Son, 1859) at 10. As such, he argued that the only legitimate restrictions on liberty could be derived from the harm principle, namely that "the only purpose for which power may be rightfully exercised over any member of a civilized community, against his will, is to prevent harm to others. His own good, either physical or moral, is not sufficient warrant.": at 9.

³⁰ Richard Chappell, "On Liberty," *Philosophy, et cetera*, 29 June 2004, <<http://www.philosophyetc.net/2004/06/on-liberty.html>> (27 June 2008). See also Lawrence McNamara, "Chapter 1: Free Speech," in Des Butler & Sharon Rodrick, *Australian Media Law*, (2nded, Lawbook Co, 2004) at 7.

³¹ Christopher Wonnell, "Truth and the Marketplace of Ideas" (1986) 19 *University of California at Davis Law Review* 669 at 671.

³² Lawrence McNamara, "Chapter 1: Free Speech," in Des Butler & Sharon Rodrick, *Australian Media Law*, (2nded, Lawbook Co, 2004) at 7.

³³ Lawrence McNamara, "Chapter 1: Free Speech," in Des Butler & Sharon Rodrick, *Australian Media Law*, (2nded, Lawbook Co, 2004) at 8.

³⁴ R. George Wright, "Why Free Speech Cases Are as Hard (And as Easy) as They Are" (2001) 68 *Tenn. L. Rev.* 335, at 339.

³⁵ See for example Christopher Wonnell, "Truth and the Marketplace of Ideas" (1986) 19 *University of California at Davis Law Review* 669 at 671-672; Frederick Schauer, *Free Speech: A Philosophical Enquiry* (1982) at 23; Stanley Ingber, "The Marketplace of Ideas: A Legitimizing Myth" [1984] *Duke Law Journal* 1.

³⁶ Christopher Wonnell, "Truth and the Marketplace of Ideas" (1986) 19 *University of California at Davis Law Review* 669 at 672-673.

³⁷ For a detailed discussion see Kent Greenawalt, "Free Speech Justifications" (1989) 89(1) *Columbia Law Review* 119 at 131-141; Christopher Wonnell, "Truth and the Marketplace of Ideas" (1986) 19 *University of California at Davis Law Review* 669 at 673-674.

Chapter 6: Why Mathematics Is Not Patentable

Autonomy

Freedom of expression is also said to enhance the autonomy of the individual both as an influence, and as a capacity.³⁸ As an influence, it acts by enabling access to information which influences individual decision making, leading to a better use of their autonomy.³⁹ Similarly, having a wide variety of information available encourages self-reflection, since it allows an individual “to question and claim assumptions that they might otherwise simply accept uncritically and thereby increase their autonomy.”⁴⁰ On a similar note, free expression may be seen to promote intellectual, moral and social capacities of individuals.⁴¹ As a capacity, expression should be understood as “something that is directly and intrinsically valuable of itself, rather than being something that is instrumentally valuable”.⁴² On this account, expression is the means by which individuals give effect to their autonomy.⁴³

For Kant, “autonomy is not only clearly recognisable as a free speech value for the sake of legally protected speech, but is also at the same time nothing less than ‘the ultimate justification of the State.’”⁴⁴ This justification of free speech was also found in the judicial development of free speech by US Supreme Court Justice Walter Brandeis. For example, in *Whitney v California*,⁴⁵ his Honour said:

Those who won our independence believed that the final end of the state was to make men free to develop their faculties ... They believe that freedom to think as you will and to speak as you think are means indispensable to the discovery and spread of political truth ... But they knew order cannot be secured merely through fear of punishment for its infraction; that it is hazardous to discourage thought, hope and imagination; that fear breeds repression; that repression breeds hate; that hate menaces stable government; that the path of safety lies in the opportunity to discuss freely supposed grievances and proposed remedies; and that the fitting remedy for evil counsels is good ones.⁴⁶

³⁸ Filimon Peonidis, “Freedom of Expression, Autonomy, and Defamation” (1998) 17(1) *Law and Philosophy* 1 at 3.

³⁹ Susan Hoffman Williams, *Truth, Autonomy, and Speech: Feminist Theory and the First Amendment* (NYU Press, 2004) at 18.

⁴⁰ Susan Hoffman Williams, *Truth, Autonomy, and Speech: Feminist Theory and the First Amendment* at 18.

⁴¹ Susan Hoffman Williams, *Truth, Autonomy, and Speech: Feminist Theory and the First Amendment* (NYU Press, 2004) at 19.

⁴² Lawrence McNamara, “Chapter 1: Free Speech,” in Des Butler & Sharon Rodrick, *Australian Media Law*, (2nded, Lawbook Co, 2004) at 6.

⁴³ Susan Hoffman Williams *Truth, Autonomy, and Speech: Feminist Theory and the First Amendment* (NYU Press 2004) at 19.

⁴⁴ R George Wright, “Why Free Speech Cases Are as Hard (And as Easy) as They Are” (2001) 68 *Tenn. L. Rev.* 335 at 341. This is because freedom forms a necessary condition for Kant’s categorical imperative, “[a]ct only according to that maxim whereby you can at the same time will that it should become a universal law”: Immanuel Kant, *Grounding for the Metaphysics of Morals*, at 421, which Kant describes as “the canon for morally estimating any of our actions”: Kant, at 424.

⁴⁵ *Whitney v California* (1927) 274 U.S. 357.

⁴⁶ *Whitney v California* (1927) 274 U.S. 357 at 375.

Democracy

The final justification for free speech is based on its being a necessary condition for democracy. This is supported by a view that speech is information, and the more information an electorate is able to access, the better the quality of democratic representation they will choose.⁴⁷ However, the requirements of a functioning democratic system need not be limited to so narrow a range of expression. Democratic governance depends on the existence of a “a robust, pluralist civil society”⁴⁸ in which citizens develop “independent spirit, self-direction, social responsibility, discursive skill, political awareness, and mutual recognition”⁴⁹ in order for democratic culture to take hold. These preconditions themselves can be translated into free speech issues. For example, any true democratic system is built around the notion of one person, one vote. Translated into the realm of expression, this requires that “there are no significant inequalities of power when people communicate”.⁵⁰ Thus “it could well be argued that the state should act to remove inequalities in the communicative process and ensure that the silenced are heard and the powerful do not prevail in public discourse.”⁵¹ This is consistent with Mill’s caveat that “[l]iberty, as a principle, has no application to any state of things anterior to the time when mankind have become capable of being improved by free and equal discussion.”⁵²

Similarly, free speech helps build a culture of participation. This participation should not be limited to participation in the electoral process, but to the creation of a democratic culture, or a “culture in which people can participate actively in the creation of cultural meanings that in turn constitute them.”⁵³ Put another way, by allowing people to participate in the creation of culture, freedom of speech imbues individuals with an expectation that they should be allowed to participate in the selection of government, thereby underscoring the notion of popular sovereignty.⁵⁴

A right of free expression based on this broader understanding of democracy extends well beyond the governmental sphere into the private sector:

[The state] must facilitate the democratization of associational and communicative frameworks to provide greater opportunities for citizen engagement and self-

⁴⁷ Lawrence McNamara, “Chapter 1: Free Speech,” in Butler D & Rodrick S, *Australian Media Law*, 2nd Edition, Lawbook Co, Pyrmont NSW, 2004, at 9.

⁴⁸ Neil Weinstock Netanel, “Copyright and a Democratic Civil Society” [1996] 106 *Yale Law Journal* 283 at 342.

⁴⁹ Neil Weinstock Netanel, “Copyright and a Democratic Civil Society” [1996] 106 *Yale Law Journal* 283 at 343.

⁵⁰ Lawrence McNamara, “Chapter 1: Free Speech,” in Butler D & Rodrick S, *Australian Media Law*, 2nd ed, Lawbook Co, 2004) at 10.

⁵¹ Lawrence McNamara, “Chapter 1: Free Speech,” in Des Butler and Sharon Rodrick, *Australian Media Law* (2nd ed, Lawbook Co, 2004) at 10. Also note the democratic requirement of equality above can also suggest that speech should be *limited* in certain situations, namely where “a democratic community agrees on a moral principle that should guide actions”: *ibid*.

⁵² John Stuart Mill *On Liberty* (2nd ed, Longmans Green Reader & Dyer, 1863) at 24.

⁵³ Jack M Balkin, “How Rights Change: Freedom of Speech in the Digital Era” (2004) 26 *Sydney Law Review* 5 at 12.

⁵⁴ Neil Weinstock Netanel, “Copyright and a Democratic Civil Society” (1996) 106 *Yale Law Journal* 283 at 343. See also Marci A Hamilton, “Power, Responsibility, and Republican Democracy” (1995) 93 *Michigan Law Review* 1539 at 1539-1540; Frank I Michelman, “Law’s Republic” (1988) 97 *Yale Law Journal* 1493 at 1500-1503; John Stuart Mill, “Chapter 2: The Criterion of a Good Form of Government” in *Considerations on Representative Government* (1861) <http://ebooks.adelaide.edu.au/m/mill/john_stuart/m645r/chapter2.html> (4 March 2011): “The most important point of excellence which any form of government can possess is to promote the virtue and intelligence of the people themselves.”

Chapter 6: Why Mathematics Is Not Patentable

government. Concomitantly, it must work to modify or eliminate social arrangements that undermine democratic citizenship while still leaving considerable room for “bottom-up” community organizing, education, and direction.⁵⁵

Thankfully, it is not necessary to proffer a view on which of these characterisations of freedom of expression is the correct one. For present purposes it is sufficient to say that they help to form a backdrop against which an understanding of the impact of a patent grant can have on others. The truth justification aligns to the notion that individual contributions to an overlapping field of exploration are to be valued for the reason that any one of them might give rise to a valuable advance. As such, any restriction on freedom ought to consider the possibility that through the award of control risks the possibility that a *better* invention might be lost, and the public interest ill served as a result. The autonomy theory makes clear that individual contributions are valuable in their own right, suggesting that the ability of individuals to act without restriction is something worth protecting, regardless of any inventive merit their entry into a particular sphere may bring about. Finally, the democratic theory of patent law underscores the fact that competition is important because it builds a culture of participation, and is as such inherently valuable in a democratic society.

4 Is freedom required in mathematics?

Having defined what is meant by freedom of thought and expression, the next task is to demonstrate how innovation in mathematics depends on mathematicians having such freedoms in order to advance their art. In the previous chapter, a number of philosophies of mathematics were introduced. In developing the role of freedom in mathematics it will be necessary to rely on the understanding developed. As such, a brief review of the theories may be useful. Platonist theory posits that the realm of mathematics is made of abstract objects which exist independently of human minds, although it is through ill-defined mental faculty that they are accessed. The truth of a theory is therefore determined by comparing the theory with an “observation” of those objects. Despite being one of the oldest theories of mathematics, it is still relied on by working mathematicians, who treat the objects of mathematics as if they exist. The rationalist school holds that the objects of mathematics are purely mental creations, and the correctness of mathematical theories is ultimately reliant on human reason. The empiricist school in contrast posited that the correctness of mathematics ultimately depended on its validation through empirical observation of the real-world. Kant’s transcendental idealism is a mixture of rationalism and empiricism, which posits that it is through the combination of observation and intuition that truth could be tested.

Following the foundational crisis of mathematics towards the end of the 19th centuries, these older philosophies developed into a three-way fight for supremacy. The logicist school, a refined version of rationalism, held that all mathematics could be subsumed within logic, and attempts were made to demonstrate how all of arithmetic could be derived using rules of logical

⁵⁵ Neil Weinstock Netanel, “Copyright and a Democratic Civil Society” [1996] 106 *Yale Law Journal* 283 at 345.

Is freedom required in mathematics?

inference from a series of accepted axioms and definitions. A related school was the formalist school, which argued that mathematics was a game played using axioms and rules like those posited by the logicians. However, the choice of logic is, on a strictly formalist account, only an arbitrary one. The intuitionist school, a variation of Kant's transcendental idealism, posited that all mathematics was based on intuition derived from experience, although intuition plays the primary role.

In the 20th century, quasi-empiricism modified the empiricist position, accepting the abstract nature of mathematics, but tying its ultimate validity to empirical observation. Fictionalism asserted a new version of formalism, suggesting that the whole of mathematics was a convenient fiction. Social constructivists agreed it was a fiction, but suggested it could be verified by agreement between mathematicians. Structuralists sought to shift the debate, by replacing mathematical objects with patterns and structures, although the dependency on platonist, empiricist, rationalist and formalist methods of verification remain.

With this potted history of mathematics in mind, a consideration of the role of thought and expression in mathematics is possible. The role of thought and expression, although acknowledged as interrelated, are again considered separately.

Thought in mathematics

The word mathematics comes from the Greek word $\mu\acute{\alpha}\theta\eta\mu\alpha$ (máthema) meaning "something learned"⁵⁶ and comes from "the same Indo-European base ... as produced English memory and mind".⁵⁷ This definition alone is suggestive of the central role of thought in mathematics. Support for such claims can be found by looking at the descriptive terminology⁵⁸ applied to the

⁵⁶ "mathematics" in John Ayto, *Word Origins: the hidden histories of English words from A to Z* (A & C Black, 2006).

⁵⁷ "mathematics" in John Ayto, *Word Origins: the hidden histories of English words from A to Z* (A & C Black, 2006).

⁵⁸ It will be recalled from Chapter 3 that a finding that the steps of a particular claim were mental steps in *Halliburton Oil Well Cementing Co v Walker* (1944) 146 F.2d 817 at 821 was supported by reference to the descriptive terms used therein. See on page 154.

Chapter 6: Why Mathematics Is Not Patentable

mathematical subject matter such as “proof”,⁵⁹ “deduction”,⁶⁰ “logic”,⁶¹ “analysis”,⁶² “conjecture”,⁶³ “hypothesis”,⁶⁴ “theory”,⁶⁵ “lemma”,⁶⁶ and mathematical activities like “reasoning”, “solving”, “abstraction”,⁶⁷ “generalisation” and “demonstration”. As would be expected then, there are explicit descriptions of mathematics as being concerned with thought. Shapiro for example is in no doubt:

The practice of mathematics is primarily a *mental* activity. To be sure, mathematicians use paper, pencils, and computers, but at least in theory these are dispensable. The mathematician’s main tool is her mind. Although the philosophies [of mathematics] are quite different from (and even incompatible with each other, they all place emphasis on this activity of mathematics, paying attention to its basis or justification.⁶⁸

⁵⁹ The dominant modern conception of proof in mathematics is as “a purely logical construct validated in purely syntactic terms”: Leo Corry, “II.6 The Development of the Idea of Proof” in Timothy Gowers et al (eds), *The Princeton Companion to Mathematics* (2008) at 140. However, Corry notes that this has not always been the case, and that such a notion of proof has limitations, and a number of “alternative conceptions of what should count as valid mathematical argument have become increasingly accepted as part of current mathematical practice.”: at 141. In a different context, Atiyah relevantly notes that “[p]roof is the end product of a long interaction between creative imagination and critical reasoning”: Sir Michael Atiyah, “VIII.6 Advice to a Young Mathematician” in Gowers et al.

⁶⁰ “When the modern subject is studied as an abstract deductive science in its own right, it is often referred to more fully as pure mathematics”: “mathematics”, *OED Online*, November 2010 <<http://www.oed.com/view/Entry/114974>> (11 September 2011).

⁶¹ “The study of numbers, shapes, and other entities by logical means.”: “mathematics” in *The Penguin Dictionary of Mathematics* (2008); “a science (or group of related sciences) dealing with the logic of quantity and shape and arrangement”: “mathematics”, *Princeton WordNet* <<http://wordnetweb.princeton.edu/perl/webwn?s=mathematics>> (1 March 2011). As noted in the previous chapter, it forms the core of the logicist theory of mathematical philosophy; as well as being a branch of mathematics. The development of logic as a branch of mathematics was relevantly described by George Boole in the Introduction to his book *An Investigation of the Laws of Thought* (Project Gutenberg, 2005) at 1: “The design of the following treatise is to investigate the fundamental laws of those operations of the mind by which reasoning is performed; to give expression to them in the symbolic language of a Calculus, and upon this foundation to establish the science of Logic and construct its method.” The role of logic, and the impact of Boole’s project on computer science are discussed in Chapter 7.

⁶² Used as both a classification of the subject matter of mathematics, but also as part of the research methodology of mathematics in any area.

⁶³ A conjecture is “[a] statement which may be true, but for which a proof (or disproof) has not been found”: “conjecture (hypothesis)” in R David Nelson (ed), *The Penguin Dictionary of Mathematics* (Penguin, 2008). An example of a conjecture is the Goldbach conjecture, which asserts that “every even number greater than 2 is the sum of two primes”: “Goldbach’s conjecture,” *The Penguin Dictionary of Mathematics* (Princeton University Press, 2008). See also “VI.17 Christian Goldbach” in Gowers et al (eds), *The Princeton Companion to Mathematics* (2008) at 745; Leo Corry, “II.6 The Development of the Idea of Proof” in Gowers et al at 142: “many branches of mathematics now contain conjectures that seem to be both fundamentally important and out of reach for the foreseeable future. Mathematicians persuaded of the truth of such conjectures increasingly undertake the systematic study of their consequences, assuming that an acceptable proof will one day appear (or at least that the conjecture is true)”.

⁶⁴ The term is used interchangeably with conjecture, an example being Riemann’s hypothesis, which does not bear simple explanation. For more see “Riemann zeta function,” in *The Penguin Dictionary of Mathematics* (2008); Andrew Granville, “IV.2 Analytic Number Theory” in Timothy Gowers et al (eds), *The Princeton Companion to Mathematics* (2008) at 337.

⁶⁵ For a specific example, see “Cantor’s theory of sets” in R David Nelson (ed), *The Penguin Dictionary of Mathematics* (Penguin, 2008). The term is often used to describe branches of “pure” mathematics, such as number theory, knot theory, game theory, decision theory and information theory.

⁶⁶ A lemma is “a subsidiary proposition that is assumed to be true in order to prove another proposition”: “lemma”, *Princeton WordNet* <<http://wordnetweb.princeton.edu/perl/webwn?s=lemma>> (1 March 2011).

⁶⁷ In relation to the central role which abstraction plays in mathematics, see Chapter 5, Section 1 on page 189.

⁶⁸ Stewart Shapiro, *Thinking About Mathematics* (Oxford University Press, 2000) at 172-3. Similarly, the definition put forward in Chapter 5 suggested mathematics is “a process of thinking that involves building and applying abstract, logically connected networks of ideas.”

Further, many of the philosophies of mathematics clearly place mental processes at the core. This is nowhere more stark than in intuitionism, which describes mathematics as the process of “deducing theorems exclusively by means of introspective construction”,⁶⁹ and thereby “a study of certain functions of the human mind”.⁷⁰ The role of thought in the rationalist school is also clear, since adherents were “impressed with the seemingly unshakeable foundation enjoyed by all mathematics, and its basis in pure rationality. They tried to put all knowledge on the same footing.”⁷¹ Similarly, Frege’s logicist programme⁷² and Kant’s transcendental idealism⁷³ also place obvious emphasis on the role of mental processes in mathematics.

Other accounts of mathematics might be less obvious in their endorsement of the primary role of the mind in mathematics, but at the very least do not contradict such a claim. Recall from Chapter 5 the formalist characterisation of mathematics as a “meaningless game”.⁷⁴ Accepting this claim in the strictest sense does not mean mental activity is not central to the formalist programme. The formulation of the rules of the system (the axioms and allowable deductions), the “playing” of the game, and the underlying goal of replacing the ambiguities of intuition in mathematics with a rigorous and coherent formal foundation all point towards the primacy of mental processes in the formalist account. If one accepts a less radical formalism, wherein it is admitted that the symbols have meanings, and the game results in logical consequences,⁷⁵ the significance of the mind is further amplified.

The platonist account, which build mathematics upon the existence of external, abstract objects which exist independently of humanity, does not on the face of it depend on thought. The independence of these objects means that they would exist whether humans think about them or not. However, thought remains at the centre of the platonist accounts, either as a means of access of the objects in question through some form of mental reflection or intuition.⁷⁶ At the

<<http://www.project2061.org/publications/bsl/online/ch9/ch9.htm#MathematicalWorld>> (accessed 15 May 2007). See Chapter 5, at 189. See also Dirk J Struik, *A Concise History of Mathematics* (Dover Publications, 1967) at 1: “Mathematics is a vast adventure in ideas; its history reflects some of the noblest thoughts of countless generations.”; Leonard Peikoff, *Objectivism: The Philosophy of Ayn Rand*, Penguin Group, New York 1991, at 90: “Mathematics is the substance of thought writ large, as the West has been told from Pythagoras to Bertrand Russell; it does provide a unique window into human nature.”; “Mathematics” in Kenneth McLeish (ed), *Bloomsbury Guide to Human Thought* (Bloomsbury Publishing, 1993): “[mathematics is] an area of thought basic to the whole of modern civilization”.

⁶⁹ LEJ Brouwer, “Consciousness, Philosophy and Mathematics” (1948) cited in Stewart Shapiro, *Thinking About Mathematics* (Oxford University Press, 2000) at 8.

⁷⁰ Arund Heyting in *Intuitionism: An Introduction* (1956), cited in Stewart Shapiro, *Thinking About Mathematics* (Oxford University Press, 2000) at 8.

⁷¹ Stewart Shapiro, *Thinking About Mathematics* (Oxford University Press, 2000) at 3.

⁷² See Chapter 3, at 203.

⁷³ See Chapter 3, at 200.

⁷⁴ See Chapter 5, at 204.

⁷⁵ This is the sense in which Bertrand and Russell put forward their formal system as a foundation for mathematics. If one accepts the axioms of the systems as true, and the rules of the game as valid, it follows as a logical consequence that any results of playing the game are also true statements.

⁷⁶ Whilst the previous chapter noted the vagaries of how the realm of platonic objects is accessed, Plato believed that these objects, “can only be seen with the eye of the mind”: Plato, “The Republic” in *The Dialogues of Plato translated into English with Analyses and Introductions by B. Jowett, M.A. in Five Volumes* (3rd ed, Oxford University Press, 1892) at line 511 (Stephanus numbering). Elsewhere, in the Meno dialogue, Plato asserts that this realm of objects is remembered from a past life: see Shapiro at 52. Similarly, Gödel is attributed with believing that mathematical objects are accessed by some form of mathematical intuition: see Kurt Gödel, “Russell’s Mathematical Logic” in 1944 in Paul Benacerraf and Hilary Putnam, *Philosophy of Mathematics* (2nd ed, Cambridge University Press, 1983) at 449, discussed in Stewart Shapiro, *Thinking About Mathematics* (Oxford University Press, 2000) at 205.

Chapter 6: Why Mathematics Is Not Patentable

very least, the independence of these objects of sensory experience is not inconsistent with the mental nature of mathematical activity noted at the outset of this section.⁷⁷

Social constructivism is, at least on the face of it, harder to reconcile with a thought requirement, to the extent that it focuses attention on the social construction of mathematics through “social acts, decisions, or practices”⁷⁸ of mathematicians. Hersh for example seems to deny the objects of mathematical practice are abstract entities, being neither mental nor physical.⁷⁹ For this reason, it may seem convenient to play down the role of thought in favour of a focus on expression in the social constructivist account, which seems to accord with the importance attached to act of sharing.⁸⁰ Apart from the mental nature of many of the practices of individual mathematicians noted above, on a social constructivist account the ability of the individual mathematical practices to centralise “around a (coherent) characterization of the structure of the domain in question”⁸¹ follows from “the objective nature of the logical tools used in the characterization and constitution of [mathematics]”.⁸² So logic, and therefore thought, lie at the heart of the structuralist account. Further, the introduction of new mathematical theories is a creative endeavour, the act of creation being a primarily mental activity.⁸³ Similarly, Field’s fictionalist account, which asserts that mathematics is no more than a useful short-cut adopted by convention, implicitly acknowledges that, like any work of fiction, mathematics is a human (and therefore mental) creation.

Perhaps the philosophy least inclined to the characterisation of mathematics might be thought to be empiricism. Yet Shapiro notes that

[s]ince mathematical knowledge seems to be based on proof, not observation, mathematics is an apparent counterexample to the main empiricist thesis. Indeed, mathematics is sometimes held up as a paradigm of *a priori* knowledge – knowledge prior to, and independent of, experience. Virtually every empiricist took the challenge of mathematics most seriously, and some of them went to great lengths to accommodate mathematics, sometimes distorting it beyond recognition (see Parsons [Mathematics in Philosophy] 1983: Essay 1)⁸⁴

⁷⁷ Plato in particular placed mathematics at the centre of his philosophy, noting that it “naturally awakens the power of thought ... to draw us towards reality”: Plato *The Republic*, at 521, cited in Stewart Shapiro, *Thinking About Mathematics* (Oxford University Press, 2000) at 62. Plato, in distinguishing the role of sensory experience notes that “the real object of the entire subject is ... knowledge .. of what eternally exists, not of anything that comes to be this or that at some time and ceases to be”: at 527a, cited in Shapiro at 56 (emphasis added).

⁷⁸ Julian C Cole, “Mathematical Domains: Social Constructs?” in Bonnie Gold and Roger Simons (eds), *Proof and Other Dilemmas: Mathematics and Philosophy* (Mathematics Association of America, 2008) at 113.

⁷⁹ See Julian C Cole, “Mathematical Domains: Social Constructs?” in Bonnie Gold and Roger Simons (eds), *Proof and Other Dilemmas: Mathematics and Philosophy* (Mathematics Association of America, 2008) at 119. It should be noted however, that Hersh relies on a restricted understanding of what it is to be abstract, sourced in a platonic conception which Cole interprets to include four features: acausality, non-spatio-temporality, eternity and changelessness. See also Cole, at 119-120.

⁸⁰ See below.

⁸¹ Julian C Cole, “Mathematical Domains: Social Constructs?” in Bonnie Gold and Roger Simons (eds), *Proof and Other Dilemmas: Mathematics and Philosophy* (Mathematics Association of America, 2008) at 117.

⁸² Julian C Cole, “Mathematical Domains: Social Constructs?” in Bonnie Gold and Roger Simons (eds), *Proof and Other Dilemmas: Mathematics and Philosophy* (Mathematics Association of America, 2008) at 118.

⁸³ Julian C Cole, “Mathematical Domains: Social Constructs?” in Bonnie Gold and Roger Simons (eds), *Proof and Other Dilemmas: Mathematics and Philosophy* (Mathematics Association of America, 2008) at 121, citing Hersh with approval. The nature of mathematics as a creative endeavour is touched upon in Section 4 on the next page below.

⁸⁴ Stewart Shapiro, *Thinking About Mathematics* (Oxford University Press, 2000) at 3.

The importance of thought to mathematics is abundantly clear in Quine's quasi-empiricist philosophy, in which mathematics forms part of a web of *belief* which is validated by empirical observation, rather than forming the subject of any direct empirical observation. Such a position is consistent with the traditional empirical view, under which mathematics could be characterised as a system of beliefs which ultimately derive their truth value from empirical observation.

It is submitted that the close relationship between mathematics and philosophy evident in the philosophical debates can also be understood independently as further evidence of the nature of mathematics as a mental activity. Apart from the philosophical accounts of mathematics just discussed, the content of mathematics itself at times raises philosophical questions. For example, the Skolem paradox raises issues about "the human ability to characterize and communicate concepts";⁸⁵ Cantor's continuum theory may have implications for our understanding the nature of truth and proof;⁸⁶ and Gödel's incompleteness theorem has implications for our understanding of the way the human mind operates.⁸⁷

Expression in mathematics

Expression in the philosophies of mathematics

Some philosophies of mathematics clearly embrace expression as a key component of mathematics. In particular, nominalist accounts such as fictionalism and social constructivism emphasise the nature of mathematics as a cultural institution. On these accounts, it is the sharing of mathematics, and the shared acceptance which that engenders, which justifies the truth of mathematical propositions. That is, in order for 'new' mathematical theories to become knowledge, their truth must be ascertained by reference to the community of mathematicians who review and criticise them. This ability to share obviously requires that these propositions be expressed in some form. As such it is entirely consistent with the 'marketplace of ideas' theory of free speech.⁸⁸

Similarly, for the formalist account, the expression of mathematics in symbolic form on paper is the ultimate manifestation of mathematics. To understand how formalism fits with free expression we have only to understand what the individual deductive steps in a formal system represent. Axioms are fundamental assumptions so evident that they can be taken at face value. Theorems are derived from axioms by a set of allowed deductions. A deduction is "a process of reasoning by which a specific conclusion necessarily follows from a set of general premises".⁸⁹ In other words, a deductive step can be thought of as a fundamental discrete unit of logical

⁸⁵ Stewart Shapiro, *Thinking About Mathematics* (Oxford University Press, 2000) at 40.

⁸⁶ See the discussion in Stewart Shapiro, *Thinking About Mathematics* (Oxford University Press, 2000) at 42-43.

⁸⁷ The incompleteness theorem is summarised below. Stewart Shapiro, *Thinking About Mathematics* (Oxford University Press, 2000) at 43-44 summarises the implications of the theorem as follows: "[s]ome philosophers take the incompleteness theorem to refute mechanism, the thesis that the human mind operates like a machine. ... Gödel himself drew the careful conclusion that either the mind is not a machine or there are arithmetic questions that are 'absolutely undecidable', questions that are unanswerable by humans even in principle." The implications for such a conclusion for our understanding of software is discussed further in the next chapter.

⁸⁸ See *Abrams v. United States*, 250 U.S. 616 (1919) per Holmes J (dissenting).

⁸⁹ "deduction" in *Collins English Dictionary and Thesaurus* (Harper Collins, 1993)

Chapter 6: Why Mathematics Is Not Patentable

thought. The purpose of formal systems is to establish the truth or falsity of mathematical hypotheses according to a strict set of rules. The importance of the formalist enterprise is that it is intended to be communicated to other mathematicians in order that the claimed 'proof' can be accepted. It is clear therefore that the formalist conception of mathematics depends on the free transmission of formal proofs, and therefore freedom of expression is required.

Intuitionism takes what is perhaps the most contrary view, in that the assertion that "[i]n its simplest form mathematics remains confined to one mind".⁹⁰ It was noted by Heyting in the discussion in Chapter 5 that mathematics "cannot be rendered exactly by means of language".⁹¹ Yet the intuitionist account does not preclude expression, merely highlights the difficulties attendant on communicating mathematics between individuals. Whilst for an intuitionist, "[m]athematical ideas are independent of the dress of language",⁹² the practice of mathematics, even for intuitionists, necessarily involves such expression.

This concern with the mental aspects of mathematics is also evident in the platonist account. As was noted above. Given the interplay between expression and thought noted by Vygotsky above,⁹³ it is clear that in any event, the need to protect freedom of thought would extend to the protection of freedom of expression, on either an intuitionistic or platonist account. This is especially the case where the net result of allowing monopolies on these ideas would conflict directly with the notion of autonomy.

Other philosophical accounts are generally silent on the role of expression in mathematics.⁹⁴ In support of the contention that mathematics requires freedom of expression then, it is apposite to note firstly the interconnectedness of thought and expression noted earlier in this chapter. Further it is submitted that the consistency of these accounts with mathematics can be justified through an exposition of the role of expression in mathematical practice.

It will be recalled that rationalists believed that truth could be accessed directly by human reason. The logicist school accords with this view, but defines human reason more narrowly as logic. It will also be recalled that truth is one of the justifications put forward for free speech. In both contexts, it is the application of human reason to presented facts which allows for their truth value to be assessed. Thus the rationalist understanding of maths is clearly consistent with a mathematics exception based on freedom of speech.

Empiricists believed that mathematics is like any other science, in that it ultimately depends for its validation on evidence that is observable by the senses. In this sense it stands in direct contrast to the rationalist school, so cannot be reconciled with freedom on the same basis. Yet

⁹⁰ Arend Heyting, "Intuitionistic views on the nature of mathematics," (1974) 27(1) *Synthese* 79 at 80.

⁹¹ Arend Heyting, "Intuitionistic views on the nature of mathematics," (1974) 27(1) *Synthese* 79, 89. See also Morris Kline, *Mathematical Thought from Ancient to Modern Times* (Oxford University Press, 1972) at 1201: "Mathematical ideas are independent of the dress of language and in fact far richer."

⁹² Morris Kline, *Mathematical Thought from Ancient to Modern Times* (Oxford University Press, 1972) at 1200-1201.

⁹³ See the discussion on page 236 above. In essence, Vygotsky noted that "[i]t is unsatisfactory and even painful to the thinker himself, if he is not permitted to communicate his thoughts to others, and it is obviously of no value to his neighbours. Moreover it is extremely difficult to hide thoughts that have any power over the mind." Lev S Vygotsky and Alex Kozulin *Thought and Language* (2nd ed, MIT Press, 1986).

⁹⁴ The remaining accounts are platonism, rationalism, empiricism, transcendental idealism, logicism, quasi-empiricism, and those structuralist accounts which depend on the aforementioned philosophies in determining the nature of the structures involved.

a variation on the above argument used for rationalism is appropriate here too, as empiricism also sees mathematics as a quest for the truth. Although it is not the process of reason by which truth is assessed, it is still to be determined according to objective criteria. Mathematical truths, once proved, then are much like laws of nature in as much as they are a correct explanation of a set of evidence.

On the empiricist approach, a mathematical innovation, advance or hypothesis, is to be accorded no weight until proved true. Up until this point then, it is nothing more than an idea. There is also a fundamentally democratic understanding shared between the two that any new theory should be freely put forward, and then tested for its truth value which mathematics then shares with science.⁹⁵ This is something that is understood in mathematics, as any grand theory, such as Hilbert's formalist approach, was open to be knocked down as it was by Gödel, regardless of which mathematician had the better reputation at the time.

Mathematics as a language

To the non-mathematician, mathematics is a swirling morass of symbols. Further, the understanding of mathematics as dominated by rigour and directed to issues of proof might suggest no room for expressivity. Thus to demonstrate the expressive nature of mathematics as a language, the similarity between mathematics and literature will be illustrated. Consider the following excerpt translation of the Russian novel Eugene Onegin by Aleksandr Pushkin, as translated by James Falen:⁹⁶

And then with verse of quickened sadness
He honored too, in tears and pain,
His parents' dust... their memory's gladness...
Alas! Upon life's furrowed plain —
A harvest brief, each generation,
By fate's mysterious dispensation,
Arises, ripens, and must fall;
Then others too must heed the call.
For thus our giddy race gains power:
It waxes, stirs, turns seething wave,
Then crowds its forebears toward the grave.
And we as well shall face that hour
When one fine day our grandsons true
Straight out of life will crowd us too!

⁹⁵ See Robert K Merton, "Science and Technology in a Democratic Order" (1942) 1 *Journal of Legal and Political Sociology* 115.

⁹⁶ This example is taken from an interview with Douglas Hofstadter, a computer scientist whose academic work on the nature of the human mind is the source of much of the author's understanding of the Gödel incompleteness theorems, and the relationship of software to mathematics (which was considered in Chapter 1). See Tal Cohen, "An Interview with Douglas Hofstadter," 11 June 2008, <http://tal.forum2.org/hofstadter_interview> (15 June 2008).

Chapter 6: Why Mathematics Is Not Patentable

Now consider a translation of the same passage by a different translator, Douglas Hofstadter:⁹⁷

And there he, on the stark, dark marker
Atop his parents' graves, shed tears,
And praised their ashes — darker, starker.
Alas, life reaps too fast its years;
All flesh is grass. Each generation,
At heaven's hidden motivation,
Arises, blooms, and falls from grace;
Another quickly takes its place.
And thus our race, rash and impetuous,
Ascends and has its day, then raves
And hastens toward ancestral graves.
All too soon, death's sting will get to us;
Aye, how our children's children rush
And push us from this world's sweet crush.

A side-by-side comparison of these examples makes clear both the similarity of the underlying concepts, and the way in which the translators have injected their own individual style into the translation. From the copyright perspective, this example aptly illustrates differences between idea and expression, and that the two are sufficiently different to warrant individual protection. Similar patterns emerge in mathematics. For example, the following is an excerpt of a famous mathematical proof, Gödel's Incompleteness Theorem, which was introduced briefly in Chapter 1, as it appeared in an English translation of Gödel's 1944 paper:

Proposition VI: To every ω -consistent recursive class c of formulae there correspond recursive class-signs r , such that neither $v \text{ Gen } r$ nor $\text{Neg } (v \text{ Gen } r)$ belongs to $\text{Flg } (c)$ (where v is the free variable of r).⁹⁸

...

Proposition XI: If c be a given recursive, consistent class of formulae, then the propositional formula which states that c is consistent is not c -provable; in particular, the consistency of P is unprovable in P , it being assumed that P is consistent (if not, of course, every statement is provable).⁹⁹

Whilst the ordinary reader might be able to identify these statements as mathematical in nature, they are likely to be largely incomprehensible beyond that, without prior knowledge or further explanation, much as Pushkin's *Eugene Onegin* in the original Russian would be largely incomprehensible to an English speaker. Putting that aside however, the meaning of these two theorems can be summarised as follows:

⁹⁷ Taken from Tal Cohen, "An Interview with Douglas Hofstadter," 11 June 2008, <http://tal.forum2.org/hofstadter_interview> (15 June 2008).

⁹⁸ Kurt Gödel, *On Formally Undecidable Propositions of Principia Mathematica and Related Systems* (B Melzer (trans), Dover Publications, 1992) at 57.

⁹⁹ Kurt Gödel, *On Formally Undecidable Propositions of Principia Mathematica and Related Systems* (B Melzer (trans), Dover Publications, 1992) at 70 (footnotes omitted).

Is freedom required in mathematics?

Any sufficiently powerful formal system cannot be both complete and consistent.¹⁰⁰

This statement is much more digestible, despite some remnants of mathematical jargon, but contains a similar idea to the following:

All we know of the truth
Is that absolute truth, such as it is,
Is beyond our reach.¹⁰¹

On this basis it is apparent that mathematics is a form of language, as impenetrable to some as Japanese or Arabic might be to others, yet with its own peculiar pleasures:

Mathematics is in the first place a language in which we discuss those parts of the real world which can be described by numbers or by similar relations of order. But with the workaday business of translating the facts onto this language there naturally goes, in those who are good at it, a pleasure in the activity itself. They find the language richer than its bare content; what is translated comes to mean less to them than the logic and the style of saying it; and from these overtones grows mathematics as a literature in its own right. Mathematics in this sense is a form of poetry, which has the same relation to the prose of practical mathematics as poetry has to prose in any other language. The element of, the delight in exploring the medium for its own sake, is an essential ingredient in the creative process.¹⁰²

Once it is accepted that mathematics is capable of expressing ideas, it might be wondered why the language of mathematics is so unlike natural language. The reason is clear from the following passage:

The main reason for using mathematical grammar is that the statements of mathematics are supposed to be completely precise, and it is not possible to achieve complete precision unless the language one uses is free of many of the vaguenesses and ambiguities of ordinary speech. Mathematical sentences can also be highly complex; if the parts that made them up were not clear and simple, then the unclarities would rapidly accumulate and render the sentences unintelligible.¹⁰³

This might be little more than a quaint observation were it not for the importance which a recharacterisation of a mathematical problem can have. In many circumstances, the way in which a problem is expressed can lead to solutions, insights, or connections with other areas

¹⁰⁰ The above simplification of the theorem is in the author's words, but is based on a number of different formulations, influenced by the following sources: Douglas Hofstadter, *Gödel, Escher, Bach: An Eternal Golden Braid* (Basic Books, 1980); "Formalism" on *Bookrags.com* <<http://www.bookrags.com/research/formalism-wom/>> (14 July 2007).

¹⁰¹ Nicholas of Cusa (1401-64) *De Docta Ignorantia*. Cited in Paola Zizzi, "Poetry of a Logical Truth" <http://www.thalesandfriends.org/en/papers/pdf/zizzi_paper.pdf> (16 June 2008).

¹⁰² Jacob Bronowski, *Science and Human Values* (Pelican, 1964) at 21.

¹⁰³ "1.2 The Language and Grammar of Mathematics" in Timothy Gowers et al (eds), *The Princeton Companion to Mathematics* (2008)

Chapter 6: Why Mathematics Is Not Patentable

of mathematics.¹⁰⁴ It is this fluidity which is required for the advance of mathematics, and which directly demonstrates both the need for freedom of expression, and the interrelationship between expression and thought.

Mathematics as an aesthetic activity

This interplay between expression and thought hints at the inadequacy of a teleological understanding of mathematics, and directs attention towards the motivational factors which drive the mathematical enterprise forward. Although the usefulness of mathematics to the sciences is not disputed, many mathematicians have describe their discipline by reference to the concept of beauty.

The interrelationship of mathematics and beauty was introduced by the Pythagoreans, for whom number was the essence of everything,¹⁰⁵ and for whom beauty corresponded with symmetry.¹⁰⁶ Similarly, Aristotle noted that “[t]hose who assert that the mathematical sciences say nothing of the beautiful are in error. The chief forms of beauty are order, commensurability, and precision.”¹⁰⁷ Thus for Aristotle, it was only by reference to mathematics that beauty could be defined.¹⁰⁸

More recently, Bertrand Russell, whose practical influence on both mathematics and the philosophy of mathematics was discussed in Chapter 5, described the beauty of mathematics as follows:

Mathematics, rightly viewed, possesses not only truth, but supreme beauty – a beauty cold and austere, like that of sculpture, without appeal to any part of our weaker nature, without the gorgeous trappings of painting or music, yet sublimely pure, and capable of a stern perfection such as only the greatest art can show.¹⁰⁹

Similarly, Hardy noted that “[t]he mathematician’s patterns, like the painter’s or the poet’s, must be *beautiful*; the ideas, like the colours or the words, must fit together in a harmonious

¹⁰⁴ See generally “I.4 General Goals of Mathematical Research” in Timothy Gowers et al (eds), *The Princeton Companion to Mathematics* (Princeton University Press, 2008).

¹⁰⁵ “The Pythagoreans went so far as to attempt to reduce all existence to numbers. There were attempts to use the characteristic concepts of number theory in biology, to predict the gestation period of animals by the use of such concepts as prime number or perfect number. The only successful Pythagorean application of this type was, of course, to music – in which the continuum of musical tones was related to the discrete domain of numbers.”: Mark Steiner, “Mathematics Applied: The Case of Addition” in Bonnie Gold and Roger Simons (eds), *Proof and Other Dilemmas: Mathematics and Philosophy* (Mathematics Association of America, 2008) 313 at 321. A modern version of this position is held by at least one prominent physicist. “[A]t the deepest level, all we find are symmetries and responses to symmetries. Matter itself dissolves, and the universe itself is revealed as one large reducible representation of the symmetry group of nature.”: Steven Weinberg, “Towards the final laws of physics,” in *Elementary particles and the laws of physics: the 1986 Dirac Memorial Lectures* (Cambridge University Press, 1989), cited in Steiner, *ibid*.

¹⁰⁶ Nathalie Sinclair, “The Roles of the Aesthetic in Mathematical Inquiry” (2004) 6(3) *Mathematical Thinking and Learning* 261 at 263.

¹⁰⁷ Aristotle, *The Metaphysics of Aristotle* (HG Bohn, 1857) XIII 3.107b.

¹⁰⁸ In recent times, Jürgen Schmidhuber has advanced an algorithmic theory of beauty. See for example Jürgen Schmidhuber, “Low Complexity Art” (1997) 30(2) *Leonardo* 97.

¹⁰⁹ Bertrand Russell, “The Study of Mathematics” in Andrew Brink, Margaret Moran and Richard A Rempel (eds), *Collected Papers of Bertrand Russell, Volume 12* (Routledge, 1985) 83 at 86.

way. Beauty is the first test: there is no permanent place in the world for ugly mathematics.”¹¹⁰ The eminent French mathematician, Henri Poincaré similarly drew attention away from the useful aspects of mathematics towards the beautiful by noting beauty as a motivational factor for mathematicians:

The mathematician does not study pure mathematics because it is useful; he studies it because he delights in it and he delights in it because it is beautiful.¹¹¹

Whilst these observations of a few mathematicians might be dismissed as elitist or frivolous, and of limited applicability to the broad practice of mathematics, they are but a few examples of a “long tradition in mathematics of describing proofs and theorems in aesthetic terms”,¹¹² and other claims by mathematicians that “their subject is more akin to an art than it is to a science, and, like the arts, [having] aesthetic goals”.¹¹³

It should also be noted at this point that what the focus is not beauty as “an objective mode of judgment used to distinguish ‘good’ from ‘not-so-good’ mathematical entities”¹¹⁴ but looking instead at “process-oriented, personal, psychological, cognitive and even sociocultural roles that the aesthetic plays in the development of mathematical knowledge”.¹¹⁵

With this in mind, it is possible to turn to specific examples of what is considered beautiful in mathematics. Whilst it is acknowledged that “[i]t may be very hard to *define* mathematical beauty, but that is just as true of beauty of any kind – we may not know quite what we mean by a beautiful poem, but that does not prevent us from recognizing one when we read it”,¹¹⁶ In order to get some sense of what mathematicians might find beautiful, consider the following example, known as Euler’s identity:

$$e^{i\pi} + 1 = 0$$

In fact, it was ranked first by readers of a mathematics journal as *the* most beautiful mathematical theorem.¹¹⁷ The reason for its asserted beauty is the unexpected relationship it documents between five fundamental mathematical constants:

- 0 — the additive identity;
- 1 — the multiplicative identity;
- π – the circular constant;

¹¹⁰ Godfrey H Hardy, *A Mathematician’s Apology* (Cambridge University Press, 1992) at 85.

¹¹¹ Jules Henri Poincaré, “Chapter 1: The Selection of Facts” *Science et méthode* (Francis Maitland (trans), 1908), at 22.

¹¹² Nathalie Sinclair, “Aesthetics as a liberating force in mathematics education?” (2009) 41 *ZDM Mathematics Education* 45, at 45.

¹¹³ Nathalie Sinclair, “Aesthetics as a liberating force in mathematics education?” (2009) 41 *ZDM* 45, at 45.

¹¹⁴ Nathalie Sinclair, “The Roles of the Aesthetic in Mathematical Inquiry” (2004) 6(3) *Mathematical Thinking and Learning* 261 at 262.

¹¹⁵ Nathalie Sinclair, “The Roles of the Aesthetic in Mathematical Inquiry” (2004) 6(3) *Mathematical Thinking and Learning* 261 at 262.

¹¹⁶ Godfrey H Hardy, *A Mathematician’s Apology* (Cambridge University Press, 1992) at 85. See also the discussion of the fine arts below, section 5 on page 256.

¹¹⁷ See David Wells, “Are these the most beautiful?” (1990) 12(3) *The Mathematical Intelligencer* 37. Eminent physicist Richard Feynman also called this “the most remarkable formula in mathematics”: “Chapter 22: Algebra” *The Feynman Lectures on Physics: Volume I* (1970) at 10.

Chapter 6: Why Mathematics Is Not Patentable

e — the base of the natural logarithms; and

i — the imaginary unit.

That these five superstar numbers should be related in so simple a manner is truly astonishing. That Euler *recognized* such a relationship is a tribute to his mathematical power.¹¹⁸

With some notion of what mathematicians consider beautiful,¹¹⁹ it is then possible to set out the ways in which these aesthetics influence mathematical practice. The discussion in the previous chapter highlights the way in which the nature of truth is sometimes seen as the key issue for the philosophy of mathematics. However, in relation to understanding the practice of mathematics (or requirements thereof), its role is limited. This is because although multitudes of theorems are proved each year, not all of them are considered significant, in that they are a useful addition to the sum of mathematical knowledge.¹²⁰ In these circumstances, other criteria must operate to filter out what is considered valuable (or likely to be valuable), and it is these criteria which are encapsulated in mathematicians' claims to beauty or elegance.

Sinclair notes three ways in which the aesthetic operates in mathematical inquiry and influences the way in which mathematics is created: in evaluation, generation and motivation. The evaluative aesthetic is "the most recognized and public of the three roles",¹²¹ and concerns "judgments about the beauty, elegance and significance of entities such as proofs and theorems".¹²² The evaluative role of the aesthetic in mathematics plays two roles in the practice of mathematics:

First, it mediates a shared set of values amongst mathematicians about which results are important enough to be retained and fortified. ... Second, the aesthetic determines the personal decisions that a mathematician makes about which results are meaningful, that is, which meet the specific qualities of mathematical ideas the mathematician values and seeks.¹²³

The generative role of the aesthetic in mathematics involves "free, orderly, aesthetic exploration"¹²⁴ of a problem domain, looking to generate "new ideas and insights that could not be derived by logical steps alone".¹²⁵ Sinclair notes the difficulty of describing the role aesthetic in the generation of mathematics, since it often operates "at a tacit or even subconscious level,

¹¹⁸ William W Dunham, *Euler: the master of us all* (The Mathematical Association of America, 1999) at 98.

¹¹⁹ For a review of the many and varied conceptions of the mathematical aesthetic, see Nathalie Sinclair, "Aesthetics as a liberating force in mathematics education?" (2009) 41 *ZDM Mathematics Education* 45, at 46-49.

¹²⁰ Nathalie Sinclair, "The Roles of the Aesthetic in Mathematical Inquiry" (2004) 6(3) *Mathematical Thinking and Learning* 261 at 265.

¹²¹ Nathalie Sinclair, "The Roles of the Aesthetic in Mathematical Inquiry" (2004) 6(3) *Mathematical Thinking and Learning* 261 at 264.

¹²² Nathalie Sinclair, "The Roles of the Aesthetic in Mathematical Inquiry" (2004) 6(3) *Mathematical Thinking and Learning* 261 at 264.

¹²³ Nathalie Sinclair, "The Roles of the Aesthetic in Mathematical Inquiry" (2004) 6(3) *Mathematical Thinking and Learning* 261 at 267.

¹²⁴ Nathalie Sinclair, "The Roles of the Aesthetic in Mathematical Inquiry" (2004) 6(3) *Mathematical Thinking and Learning* 261 at 272.

¹²⁵ Nathalie Sinclair, "The Roles of the Aesthetic in Mathematical Inquiry" (2004) 6(3) *Mathematical Thinking and Learning* 261 at 264.

and intertwined as it frequently is with intuitive modes".¹²⁶ Nonetheless, this generative role is evident in the accounts of successful mathematicians.¹²⁷

Mathematicians are also specifically motivated to work in particular areas by aesthetic considerations, such as visual appeal, a sense of surprise and paradox, and social aesthetics.¹²⁸ Aesthetic considerations have both a selective function, assisting the mathematician to determine which areas to become involved in,¹²⁹ and also a heuristic function, in that they influences "the discernment of features in a situation, and thereby directing the thought patterns of the inquirer."¹³⁰

5 Freedom, the useful arts and non-patentability

It has been demonstrated how mathematics requires freedom of thought and freedom of expression. The question which then arises is how to give effect to such a requirement within patent law (or alternatively, to suggest how such a requirement is already a part of patent law). The difficulties in reconciling the legal status of mathematics in patent law with the nature of mathematics as espoused in the philosophies of mathematics was touched upon in Chapter 5.

It is clear that mathematics is a logical tool which is often used for specific functional purposes, in that it is often used in the sciences to describe artefacts of the natural world, or as a means of providing a method of calculating useful relationships. Yet the above discussion makes it clear that mathematics is also an intangible, aesthetic and expressive discipline. It is submitted that it is these characteristics which are largely explanatory of why mathematics has been excepted from patents.

It is submitted that the best vehicle for carrying the concept of freedom into patent law considerations is the distinction between the fine arts and the useful arts. In other words, it is asserted that mathematics is predominantly a fine art, rather than a useful art.

Defining the fine arts and useful arts

To understand the distinction between the fine and useful arts, it is first necessary to explore what such terms might mean. The clues to that meaning are largely historical and functional,

¹²⁶ Nathalie Sinclair, "The Roles of the Aesthetic in Mathematical Inquiry" (2004) 6(3) *Mathematical Thinking and Learning* 261 at 270.

¹²⁷ See in particular, Henri Poincaré, "Mathematical creation" in James R Newman (ed), *The world of mathematics* (Simon & Schuster, 1956) at 2041, cited in Nathalie Sinclair, "The Roles of the Aesthetic in Mathematical Inquiry" (2004) 6(3) *Mathematical Thinking and Learning* 261 at 270. The importance of such indirect approaches to gaining insight is also particularly evident in the accounts of Sir Michael Atiyah, Béla Bollobás, and Alain Connes in "VIII.6 Advice to a Young Mathematician" in Timothy Gowers et al (eds), *The Princeton Companion to Mathematics* (Princeton University Press, 2008).

¹²⁸ Nathalie Sinclair, "The Roles of the Aesthetic in Mathematical Inquiry" (2004) 6(3) *Mathematical Thinking and Learning* 261 at 275.

¹²⁹ "A mathematician has a great variety of fields to choose from, differing from one another widely in character, style, aims, and influence; and within each field, a great variety of problems and phenomena. Thus, mathematicians must select in terms of the research they pursue, the classes they teach, and the canon they pass on.": Nathalie Sinclair, "The Roles of the Aesthetic in Mathematical Inquiry" (2004) 6(3) *Mathematical Thinking and Learning* 261 at 274.

¹³⁰ Nathalie Sinclair, "The Roles of the Aesthetic in Mathematical Inquiry" (2004) 6(3) *Mathematical Thinking and Learning* 261 at 277.

Chapter 6: Why Mathematics Is Not Patentable

rather than textual, although looking at the definitions of those words can assist in reinforcing those clues.

What are the fine arts?

Historically, the traditional fine arts included painting, sculpture, drawing and architecture, although the Encyclopedia Britannica provides a broader listing:

Traditional categories in the [fine] arts include literature (including poetry, drama, story, and so on), the visual arts (painting, drawing, sculpture, etc.), the graphic arts (painting, drawing, design, and other forms expressed on flat surfaces), the plastic arts (sculpture, modeling), the decorative arts (enamelwork, furniture design, mosaic, etc.), the performing arts (theatre, dance, music), music (as composition), and architecture (often including interior design).¹³¹

What such endeavours have in common is that they “seek expression through beautiful or significant modes”.¹³² From this definition it is possible to unpack two aspects, the first being the expressive nature of such arts, the second being their reliance on beauty.¹³³ The importance given to expression underlies the fact that the value which is to be attributed to the arts does not lie in utilitarian assessments of the functional work of the created objects. Rather, the fine arts are concerned with the “creation of objects and presentations of imagination for their own sake without relation to utility.”¹³⁴

There is no doubt that some of these arts, most obviously architecture, can contain functional elements, although it is the beautiful, or aesthetic qualities which are the predominant focus.¹³⁵ Reliance on beauty may be seen as problematic by some. Socrates for example said that “all beauty is difficult”.¹³⁶ More recently, it has been pointed out that

¹³¹ Encyclopaedia Britannica, “the arts” *Encyclopaedia Britannica Online* <<http://www.britannica.com/EBchecked/topic/36405/the-arts>> (10 August 2011).

¹³² “fine arts,” *The Macquarie Dictionary*, (4th ed, 2005). Other definitions also note the importance of beauty, or aesthetics. “Art that is produced more for beauty or spiritual significance than for physical utility.”: “fine arts,” Eric D Hirsch, Joseph F Kett and James S. Trefil (eds), *The New Dictionary of Cultural Literacy* (Houghton Mifflin, 2002). The importance of beauty is also reflected in the alternative names given to the fine arts, namely the belle arts or beaux arts.

¹³³ The word significant will be purposefully avoided, as its meaning in the artistic context is different from the way it is commonly used in law, and may cause misunderstanding. However, it is apposite to note some expanded definitions: “the creation of works of beauty or other special significance” (emphasis added): “art,” in *Collins English Dictionary* (Collins, 2000); “Art that is produced more for beauty or spiritual significance than for physical utility.” (emphasis added): “fine arts,” in Eric D Hirsch, Joseph F Kett and James S. Trefil (eds), *The New Dictionary of Cultural Literacy* (Houghton Mifflin, 2002).

¹³⁴ Robert I Coulter, “The Field of the Statutory Useful Arts: Part II” (1952) 34 *Journal of the Patent Office Society* 487 at 494. This point of contrast between fine and useful arts is explored further below.

¹³⁵ “Arts judged predominantly in aesthetic rather than functional terms, for example painting, sculpture, and print making. Architecture is also classified as one of the fine arts, though here the functional element is also important.” “fine arts” in *The Hutchinson Unabridged Encyclopedia including Atlas*, (Helicon, 2005). It has also been noted that since the mid-18th century, “the concept of ‘utilitarianism’ (functionality or usefulness) was used to distinguish the more noble ‘fine arts’ (art for art’s sake), like painting and sculpture, from the lesser forms of ‘applied art’, such as crafts and commercial design work, and the ornamental ‘decorative arts’, like textile design and interior design.”: “Definition of Art” <<http://www.visual-arts-cork.com/art-definition>> (25 March 2011).

¹³⁶ Howard Caygill, *A Kant Dictionary* (Blackwell Reference, 1995) at 91.

terms such as beautiful and ugly seem too vague in their application and too subjective in their meaning to divide the world successfully into those things that do, and those that do not, exemplify them. Almost anything might be seen as beautiful by someone or from some point of view; and different people apply the word to quite disparate objects for reasons that often seem to have little or nothing in common.¹³⁷

Even in the visual arts, in the wake of the cubist and dadaist movements, the traditional notion of beauty as the aim of art has come under prolonged attack, to the point where “[b]eauty had disappeared not only from the advanced art of the 1960’s but from the advanced philosophy of art of that decade as well.”¹³⁸ Still, the concept has its uses. Firstly, as noted above,¹³⁹ it is by reference to the concept of beauty that mathematicians have often described their work. Secondly, an understanding of artistic works as beautiful draws attention to the traditional fine arts concern with emotional responses. From this, it is possible to broaden the label to the notion of the aesthetic, which encompasses not only beauty.

The liberal arts Although the distinction in Australia is drawn between fine arts and useful arts,¹⁴⁰ in the US the nomenclature ‘liberal arts’ is often used instead, and the definitions of the two terms suggest some difference in focus. Coulter also adds a third category which lie beyond the useful arts, which includes, business teaching and politics, referring collectively to the three groups as the ‘cultural arts’.¹⁴¹ Even allowing for a significant overlap between the three categories, this might be thought to create a source of ambiguity. It may also be an unimportant point, to the extent that the focus for the purposes of determining the boundaries of patent law is on whether something is a useful art or not. To that end, the description “fine arts” is probably better understood in the broad sense, that is, as all which is not a useful art. However, for the purposes of completeness, the content of the liberal arts is set out below.

The origin of the term ‘liberal arts’ is an interesting topic in its own right, and has much in common with the history of mathematics detailed in the last chapter. The term can be traced back to antiquity, and in particular Plato, for whom they represented a path of learning to be traversed by his Guardians in order to prepare for the study of philosophy and develop the wisdom required to rule over society.¹⁴² Plato’s scheme came to form the basis of the teachings of medieval universities. The word liberal is important in that it denotes the higher nature of the learning involved, the liberal arts forming the subject of “aristocratic pursuits and skills, as opposed to ‘the mechanical arts’”.¹⁴³ Put another way, the liberal arts “serve the purpose of

¹³⁷ “aesthetics” in *Encyclopædia Britannica* (Encyclopædia Britannica, 2011)

¹³⁸ Arthur Danto, *The Abuse of Beauty: Aesthetics and the Concept of Art* (Open Court, 2003) at 25. It is interesting to note that the cubists in particular were influenced by mathematics, in particular the development of non-Euclidean geometries, and the notion of a fourth dimension. See for example, Tony Robbin, *Shadows of reality: the fourth dimension in relativity, cubism, and modern thought* (Yale University Press, 2006), especially Chapter 3.

¹³⁹ See Section 4 on page 252.

¹⁴⁰ *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 275.

¹⁴¹ See Robert I Coulter, “The Field of the Statutory Useful Arts: Part II” (1952) 34 *Journal of the Patent Office Society* 487. Coulter’s account is discussed in further detail below.

¹⁴² Plato, “The Republic” in *The Dialogues of Plato translated into English with Analyses and Introductions by B. Jowett, M.A. in Five Volumes* (3rd ed, Oxford University Press, 1892), Book VII.

¹⁴³ “Liberal Arts and Liberal Sciences” in Kenneth McLeish (ed), *Bloomsbury Guide to Human Thought* (Bloomsbury Publishing, 1993).

Chapter 6: Why Mathematics Is Not Patentable

training the free man, in contrast with the [mechanical or useful arts], which are pursued for economic purposes".¹⁴⁴

The seven liberal arts are traditionally broken down into two groups, the foundational *trivium*, of grammar, logic, rhetoric, and the higher *quadrivium* of arithmetic, geometry, music and astronomy. The former is concerned with the "thorough use of language in a scientific and logical way",¹⁴⁵ whilst the latter is concerned primarily with mathematical subject matter, and is directed to training the mind. Although the mathematical nature of arithmetic and geometry is obvious, music and astronomy were considered to be the application of mathematical concepts to real world phenomena.¹⁴⁶

The notion of the liberal arts as concerned with 'higher learning' is consistent with an intuitive understanding of the nature of the fine arts. The substantial overlap is demonstrated by reference to Kant's categorisation of the fine arts according to

the three ways in which human beings communicate with each other: through speech, gesture and tone. The arts of speech are rhetoric and poetry, those of gesture (or the 'formative arts') include the plastic arts of architecture and sculpture and the art of painting, while the tonal arts include those of music and colour. He also admits of mixed arts. The key to understanding these divisions is to remember that they refer to skills or practices and not to objects.¹⁴⁷

It is clear then that any definitional difficulty is due to the dual way in which the phrase fine arts is used. Firstly, there is the narrow sense, where it accords with Kant's formative arts. Secondly there is the broader sense which also includes the arts of speech and the tonal arts. It is this broader interpretation which is most suitable for present purposes. Kant's categorisation is also of interest in the present context since it draws attention to the communicative aspect of the fine arts. When the fine arts are seen as a communication, or expression, from artist to audience, the importance of freedom of expression is underscored.

What are the useful arts?

As with fine arts, it is important as a starting point to define what is meant by the "useful arts". In the US the constitution allows Congress to pass laws to promote the progress of use-

¹⁴⁴ Otto Willmann, "The Seven Liberal Arts" in *The Catholic Encyclopedia*, Volume 1. (Robert Appleton Company, 1907) <<http://www.newadvent.org/cathen/01760a.htm>> (31 Mar 2011). See also Howard Caygill, *A Kant Dictionary*, (Wiley-Blackwell, 2008) at 85: "[Aquinas] distinguished between the servile arts 'ordered to works done by the body' and those of the soul or 'liberal arts'."

¹⁴⁵ "A reading list in the 'classics,' from ancient to modern" <<http://triviumquadrivium.wordpress.com/2010/02/13/is-it-worth-reading-the-classics-if-so-what-should-i-read/>> (30 March 2011).

¹⁴⁶ This reflects the Pythagorean influence "that mathematics offered a key to understanding reality, whether this reality was conceived to have an underlying geometrical structure ... or whether it was simply seen as ordered and 'in proportion.'": Serafina Cuomo, "VI.1 Pythagoras" in Timothy Gowers et al (eds), *The Princeton Companion to Mathematics* (Princeton University Press, 2008). Astronomy was concerned with building a "geometrical model of the cosmos": Ian Mueller, "MATHEMATICS, Earlier Greek" in Donald J Zeyl, Daniel Devereux and Phillip Mitsis (eds), *Encyclopedia of Classical Philosophy* (Greenwood Press, 1997) and music, or harmonics, was concerned with such things as "numerical expression of the fundamental musical concords": Mueller, *ibid*.

¹⁴⁷ Howard Caygill, *A Kant Dictionary* (Wiley-Blackwell, 2008) at 86.

ful arts.¹⁴⁸ It is clear then that in this jurisdiction, any attempt to go beyond the bounds of “useful arts” is invalid because it is unconstitutional — a strong argument against patentability indeed.¹⁴⁹ A contrast between the non-patentable fine and patentable useful arts was also drawn in *NRDC*.¹⁵⁰ It will be shown how the definition of this term has a clear influence on the scope of patent law in Europe as well.

In 1952, Coulter provided what is “still the most exhaustive and deeply considered”¹⁵¹ attempt at defining the useful arts in the patent law context.¹⁵² Coulter’s analysis depended on both an exploration of the classical notion of the useful arts, as well as consideration of the intention of the Framers of the US Constitution.

It has already been noted that the useful arts were typically contrasted with the fine arts, or the liberal arts. The contrast against the liberal arts is particularly noteworthy, since it underlies perhaps the oldest understanding of the distinction in classical times. The liberal arts were the areas of study reserved to free men, whereas the useful, or manual, arts, were those works carried out by slaves. This distinction was carried through to the Renaissance, wherein the liberal arts formed the curriculum at medieval universities – the province of the aristocracy. In contrast, the practitioners of the manual or useful arts:

did not require a high degree of intellectual attainment and cultural education, and rarely possessed them, and they engaged in manual labor, which accounted in part for their more or less lowly social and economic position in the English class structure.¹⁵³

Built around these manual and useful arts were the guilds, through which these arts were taught by serving as apprentices.¹⁵⁴ It is at least arguable that this understanding of the useful arts is reflected in the *Statute of Monopolies*,¹⁵⁵ wherein the period of a patent corresponding in early times to the length of 2 apprenticeships.

¹⁴⁸ “The Congress shall have the power ... [t]o promote the Progress of ... useful Arts, by securing for limited Times to ... Inventors the exclusive Right to their ... Discoveries”: *Constitution of the United States of America*, art. 1, §8, cl. 8.

¹⁴⁹ “The clause is both a grant of power and a limitation. This qualified authority, unlike the power often exercised in the sixteenth and seventeenth centuries by the English Crown, is limited to the promotion of advances in the ‘useful arts.’ ... The Congress in the exercise of the patent power may not overreach the restraints imposed by the stated constitutional purpose.”: *Graham v John Deere Co* 383 US 1 (1966) at 5. As an aside it is also interesting to note the balancing of constitutional rights in this context between a right to legislate to promote the progress of useful arts on the one hand, and the constitutional limitations on restricting freedom imposed by the First Amendment.

¹⁵⁰ *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 (‘NRDC’).

¹⁵¹ Alan L Durham, “‘Useful Arts’ in the Information Age” (1999) *Brigham Young University Law Review* 1419 at 1437.

¹⁵² Robert I Coulter, “The Field of the Statutory Useful Arts: Part II” (1952) 34 *Journal of the Patent Office Society* 487. Coulter’s analysis formed part of a 3 part series of articles in which he sought to criticize the failings of the mental steps doctrine. See also Robert I Coulter, “The Field of the Statutory Useful Arts: Part I” (1952) 34 *Journal of the Patent Office Society* 426; Robert I Coulter, “The Field of the Statutory Useful Arts: Part III” (1952) 34 *Journal of the Patent Office Society* 718.

¹⁵³ Robert I Coulter, “The Field of the Statutory Useful Arts: Part II” (1952) 34 *Journal of the Patent Office Society* 487 at 496.

¹⁵⁴ Robert I Coulter, “The Field of the Statutory Useful Arts: Part II” (1952) 34 *Journal of the Patent Office Society* 487 at 496.

¹⁵⁵ *Statute of Monopolies* 1623, 21 Jac 1, c 3 (UK).

Chapter 6: Why Mathematics Is Not Patentable

Coulter's analysis concludes with a claim that the "fundamental attribute"¹⁵⁶ of the useful arts was that "[t]hey relate to "controlling the forces and materials of nature and putting them to work in a practical way for utilitarian ends serving mankind's physical welfare."¹⁵⁷ Coulter then asserts that the phrase useful arts has a contemporary equivalent — technology.¹⁵⁸ Coulter's analysis seems to have been influential on judicial development of patent law at the time.¹⁵⁹

Understood as a synonym for technology, it is possible to potentially unify the scope of patent law across the three jurisdictions thus far considered. It is apposite to note the requirement in Article 27 of TRIPS,¹⁶⁰ that members of the World Trade Organisation make patents available in all fields of *technology*, supports this unified understanding. In the EU, the link between the technological arts, and the requirement in of a technical contribution, is obvious. Even putting aside the use of the term "useful arts" in *NRDC* as noted above, it is at least arguable that the historical meaning of "manner of manufacture" in the *Statute of Monopolies* encompassed "technology" as it existed at that time. The subsequent expansion of the meaning of manner of manufacture beyond a literal interpretation might be understood as a reflection of the changing nature of technology over the subsequent period.

Defining technology But can technology be properly defined, or is this fool's errand? Durham suggests that "the more one looks at how 'technology' has been defined by scholars, the less one is sure what it means."¹⁶¹ In *CFPH's Application*,¹⁶² Deputy Judge Prescott described the related term "technical" as "a restatement of the problem in different and more imprecise language ... It is a useful servant but a dangerous master".¹⁶³ Similarly, the courts in the US, whilst toying with a technological arts requirement, eventually resiled from such an understanding of the boundary of patent law in *Lundgren*.¹⁶⁴ In the EU, the technological contribution test has been criticised as having "all the disadvantages of the original obscure wording [of Article 52], with the added disadvantage of not even providing the actual legislative test."¹⁶⁵ Similarly in *Grant*, the Full Federal Court rejected the recharacterisation of the boundaries of patent law by reference to a science or technology requirement as follows:

¹⁵⁶ Robert I Coulter, "The Field of the Statutory Useful Arts: Part II" (1952) 34 *Journal of the Patent Office Society* 487 at 498.

¹⁵⁷ Robert I Coulter, "The Field of the Statutory Useful Arts: Part II" (1952) 34 *Journal of the Patent Office Society* 487 at 498.

¹⁵⁸ "Probably the best word in common usage today that expresses this idea is 'technology'." Robert I Coulter, "The Field of the Statutory Useful Arts: Part II" (1952) 34 *Journal of the Patent Office Society* 487 at 498.

¹⁵⁹ See for example, *In re Musgrave* 431 F.2d 882 (1970) at 893: "All that is necessary, in our view, to make a sequence of operational steps a statutory 'process', within 35 USC §101 is that it be in the technological arts so as to be in consonance with the Constitutional purpose to promote the progress of 'useful arts.'"; *In re Waldbaum* 457 F.2d 997 (1972) at 1003: "The phrase 'technological arts,' as we have used it, is synonymous with the phrase 'useful arts' as it appears in Article I, Section 8 of the Constitution."

¹⁶⁰ *Agreement on Trade-Related Aspects of Intellectual Property Rights*, opened for signature 15 April 1994, 1869 UNTS 299, 33 ILM 1197 (entered into force 1 January 1995).

¹⁶¹ Alan L Durham, "'Useful Arts' in the Information Age" (1999) *Brigham Young University Law Review* 1419 at 1444.

¹⁶² *CFPH LLC's Application* [2005] EWHC 1589 (Pat).

¹⁶³ *CFPH LLC's Application* [2005] EWHC 1589 (Pat) at [14]. See the discussion at [11]-[14]. See also *Symbian Ltd v Comptroller-General of Patents* [2008] EWCA Civ 1066; [2009] RPC 1 at [30]-[32], discussed in Chapter 3 at 136in which their Honours noted that the concept "could easily mean different things to different people": at [30].

¹⁶⁴ *Ex parte Lundgren*. Appeal No 2003-2088 (2005).

¹⁶⁵ *Symbian Ltd v Comptroller-General of Patents* [2008] EWCA Civ 1066; [2009] RPC 1 at [30].

What is or is not to be described as science or technology may present difficult questions now, let alone in a future which is as excitingly unpredictable now as it was in 1623 or 1959, if not more so. We think that to erect a requirement that an alleged invention be within the area of science and technology would be to risk the very kind of rigidity which the High Court warned against.¹⁶⁶

Despite this, it is submitted that the philosophy of technology, and the definitions it offers, are of direct benefit in this context. As will be seen below, many of the definitions put forward bear a remarkable resemblance to the ways in which the patentable subject matter has been defined. As such, they offer potentially relevant insights into the debate, whilst also serving to demonstrate that the choice between various definitions is in a real sense unavoidable. In forming and reforming the approach to determining patentable subject matter in the cases that come before them, judges are in effect making implicit decisions referable to the view they have taken of technology, whether broad or narrow. By bringing these issues to the fore, it is suggested that a coherent conception of the scope of patentable subject matter might be developed.¹⁶⁷

The advantage of discussing philosophies of technology, and making an enunciated choice as to which is the appropriate one, has the advantage that it brings policy choices out into the open. When policy is swept under the carpet, it is difficult to review, and sees the patentable subject matter issue either collapse under its own weight, being reduced to nothing other than a form requirement, or sees ad hoc bouncing from under-protection to over-protection. Fear of the future development of our concept of technology stands to drive the development of an ever broader notion of what is patentable without clear conceptual guidance.

Whilst a comprehensive analysis of the multitudinous interpretations of technology is a mammoth task, a brief survey of possible alternatives is warranted.¹⁶⁸ Building on the obvious relationship between science and technology, some have suggested technology as applied science.¹⁶⁹ This view finds a modicum of support in the case law which distinguishes between knowledge and its application to some end. However, such a definition may be criticised on the basis that technology, although underpinned by science, is not predicated upon it. To put it another way, "[t]echnology is purposive and tends ... to be positivist. The criterion is simply, does it work."¹⁷⁰

¹⁶⁶ *Grant v Commissioner of Patents* (2006) 154 FCR 62 at 71.

¹⁶⁷ "As the principal legal response to technological change, the regime of patents too has suffered from its inability to develop a coherent sense of its own subject matter": John R Thomas, "The Post-Industrial Patent System," (1999) 10(3) *Fordham Intellectual Property Media and Entertainment Law Journal* 3 at 41.

¹⁶⁸ For a comprehensive account of the philosophy of technology, see Carl Mitcham, *Thinking through Technology: The Path Between Engineering and Philosophy* (University of Chicago Press, 1994). For a survey of alternatives from the perspective of their relevance to patent law, see Alan L Durham, "'Useful Arts' in the Information Age" (1999) *Brigham Young University Law Review* 1419; John R Thomas, "The Post-Industrial Patent System," (1999) 10(3) *Fordham Intellectual Property Media and Entertainment Law Journal* 3.

¹⁶⁹ Alan L Durham, "'Useful Arts' in the Information Age" (1999) *Brigham Young University Law Review* 1419 at 1445.

¹⁷⁰ Donald Cardwell, *The Norton History of Technology*, (1995) at 492-3, cited in Alan L Durham, "'Useful Arts' in the Information Age" (1999) *Brigham Young University Law Review* 1419 at 1445.

Chapter 6: Why Mathematics Is Not Patentable

A variant on that definition emphasises the role of technology in the production of what is useful or practical.¹⁷¹ This would be most consistent with the “useful result” test of *Alappat*¹⁷² and *State Street*,¹⁷³ and accords, literally speaking, with the phrase “useful arts”. However, this is a potentially broad definition, since all human endeavours aim at some ultimate good.¹⁷⁴ Despite the fact that such a distinction does offer a potential basis to distinguish the useful from the fine arts in that the former are created for a purpose, whilst the latter are ends in themselves.¹⁷⁵ The danger is that such a distinction will be overlooked, a concern which is justified by the broad reach of patent law in the wake of *State Street*.¹⁷⁶

A broad understanding put forward of the nature of technology is that which describes technology as concerned with “human work”,¹⁷⁷ or “the systematic treatment of any thing or subject”.¹⁷⁸ Such definitions flow from the central purpose which the design process plays in the development of technologies,¹⁷⁹ design being “subject to rational scrutiny but in which creativity is considered to play an important role”.¹⁸⁰ The basis of technology in rational thinking, as opposed to the “free play” of aesthetic thinking seems to offer some differentiation between the useful and fine arts respectively. Not all scholars agree with such a distinction however, with Agassi, for example, claiming that “what we call the [fine] arts ... [are] all technology”.¹⁸¹

As such, there needs to be some form of limitation on the scope of patentability. Adopting a narrower interpretation of technology would limit patent law to the “making of *physical artefacts* and the physical alteration of the environment”.¹⁸² Mitcham posits that despite the divergence of meanings attributed to technology, all conceptions all admit of a “primacy of reference to the making of material artefacts”.¹⁸³ Hannay and McGinn also highlight physicality as a way of

¹⁷¹ See Frederick Ferré, *Philosophy of Technology* (Prentice Hall, 1988) at 26, who describes technology as “practical implementations of intelligence.” Cited in John R Thomas, “The Post-Industrial Patent System” (1999) 10(3) *Fordham Intellectual Property, Media and Entertainment Law Journal* 3 at 39. Discussed in Carl Mitcham, *Thinking through Technology: The Path Between Engineering and Philosophy* (University of Chicago Press, 1994) at 156-157.

¹⁷² *In re Alappat* 33 F.3d 1526 (1994).

¹⁷³ *State Street Bank & Trust Co v Signature Financial Group Inc* 149 F.3d 1368 (1998).

¹⁷⁴ Aristotle, *Nicomachean Ethics*, (Sir David Ross (trans), Oxford University Press, 1966) at 1. Cited in Alan L Durham, “‘Useful Arts’ in the Information Age” (1999) *Brigham Young University Law Review* 1419 at 1446.

¹⁷⁵ Carl Mitcham, *Thinking through Technology: The Path Between Engineering and Philosophy* (University of Chicago Press, 1994) at 156.

¹⁷⁶ Kappos et al note that in the wake of *State Street*, patents were issued in “architecture, athletics, insurance, painting, psychology and the law itself”: David J Kappos, John R Thomas and Randall J Bluestone, “A Technological Contribution Requirement for Patentable Subject Matter: Supreme Court Precedent and Policy” (2008) 6(2) *Northwestern Journal of Technology and Intellectual Property* 152, at 164.

¹⁷⁷ Peter F Drucker, *Technology, Management and Society* (1970) at 45-46, cited in Alan L Durham, “‘Useful Arts’ in the Information Age” (1999) *Brigham Young University Law Review* 1419 at 1448.

¹⁷⁸ Charles Singer et al, “Preface” in Charles Singer et al (eds) *A History of Technology* (1954) at vii, cited in Alan L Durham, “‘Useful Arts’ in the Information Age” (1999) *Brigham Young University Law Review* 1419 at 1449.

¹⁷⁹ Maarten Franssen, Gert-Jan Lokhorst and Ibo van den Poel, “Philosophy of Technology,” *Stanford Encyclopedia of Philosophy*, <<http://plato.stanford.edu/entries/technology/>> (1 May 2011) at §2.3.

¹⁸⁰ Maarten Franssen, Gert-Jan Lokhorst and Ibo van den Poel, “Philosophy of Technology,” *Stanford Encyclopedia of Philosophy*, <<http://plato.stanford.edu/entries/technology/>> (1 May 2011) at §2.3.

¹⁸¹ Joseph Agassi, *Technology: Philosophical and Social Aspects* (1985) at 90. Cited in Alan L Durham, “‘Useful Arts’ in the Information Age” (1999) *Brigham Young University Law Review* 1419 at 1449-1450. See also Durham, n210 below and accompanying text.

¹⁸² Alan L Durham, “‘Useful Arts’ in the Information Age” (1999) *Brigham Young University Law Review* 1419 at 1447.

¹⁸³ Carl Mitcham, *Thinking through Technology: The Path Between Engineering and Philosophy* (University of Chicago Press, 1994) at 152. Mitcham’s definition affords a basis for distinguishing between human making and “human doing — for example, political, moral, religious, and related activities”: at 153. See also Maarten Franssen, Gert-Jan Lokhorst and Ibo van den Poel, “Philosophy of Technology,” *Stanford Encyclopedia of Philosophy*, <<http://plato.stanford.edu/entries/technology/>> (1 May 2011) at §2.3: “technology is a practice focused on the creation of artifacts and, of increasing importance, artifact-based services.”

distinguishing technology from other forms of human activity:

[T]echnology differs from other activity-forms in that the natural environment – both in respect to the meteorological and creature-related threats it poses to human survival, and the spatiotemporal obstacles it presents to human desires for communication and transport — is a factor that more powerfully and more directly conditions technology than is the case with other cultural forms, for example, religion and art.¹⁸⁴

Why physicality is important The idea that to be patentable a process must be “embodied and connected with corporeal substances”¹⁸⁵ extends back through patent law at least as far as *Boulton v Bull*.¹⁸⁶ It is the inherently complex relationship between the definition of a process and the physical world which has caused problems for this class of invention since the Court split evenly on the point in that same case. Despite the broadening conception of patentable subject matter since that time, physicality remains an important limiting device in the subject matter inquiry.

There is certainly some support for a broad view of the scope of patent law, as is evident from the following:

Whatever may be left of the [Freeman-Walter-Abele] test, if anything, this type of physical limitations analysis seems of little value because... “after *Diehr* and *Alappat*, the mere fact that a claimed invention involves inputting numbers, calculating numbers, outputting numbers, and storing numbers, in and of itself, would not render it nonstatutory subject matter, unless, of course, its operation does not produce ‘a useful, concrete and tangible result.’” [*State Street Bank & Trust v. Signature Financial Group*, 149 F.3d 1368] at 1374 (quoting [*In re*] *Alappat*, 33 F.3d at 1544).¹⁸⁷

This “useful result” test, which for a time was in the ascendancy in both Australia and the US,¹⁸⁸ does not equate usefulness with physicality. However, as Breyer J noted in *Bilski v Kappos*, although the scope of patentable subject matter “is broad, it is not without limit”.¹⁸⁹ Whilst not wholeheartedly adopting a physicality-based understanding of the reach of patent law, by

¹⁸⁴ See also Robert E McGinn “What is Technology?” in Paul T Durbin, (ed.) *Research in Philosophy and Technology*, Vol. I (JAI Press, 1978) at 190, describing technology as “a form of activity that is *fabricative, material product making or object transforming*, purposive (with the general purpose of expanding the realm of the humanly possible), knowledge-based, resource-employing, methodical, embedded in a socio-cultural-environmental influence field, and informed by its practitioners’ mental sets.” (emphasis added).

¹⁸⁵ *Boulton v Bull* (1795) 126 ER 651 at 667.

¹⁸⁶ Discussed in Chapter 2, section 3 on page 42.

¹⁸⁷ *AT&T v Excel Communications* 172 F.3d 1352 (1999) at 1359.

¹⁸⁸ This was the approach adopted in the United States in: *In re Alappat* 33 F.3d 1526 (Fed Cir, 1994), discussed in Chapter 3, Section 3 on page 93; *State Street Bank & Trust Co v Signature Financial Group Inc* 149 F.3d 1368 (Fed Cir, 1998) discussed in Chapter 3, at Section 3 on page 94; and *Ex parte Lundgren* Appeal No 2003-2088 (BPAI, 2005) discussed in Chapter 3, at Section 3 on page 96. This approach was endorsed in Australia in: *IBM v Commissioner of Patents* (1991) 33 FCR 218, discussed in Chapter 3, Section 3 on page 121; *CCOM Pty Ltd v Jieying Pty Ltd* (1994) 51 FCR 260, discussed in Chapter 3 at Section 123, Section and *Welcome Real-Time SA v Catuity Inc and Ors* (2001) 113 FCR 110, discussed in Chapter 3, Section 3 on page 125.

¹⁸⁹ *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3258.

Chapter 6: Why Mathematics Is Not Patentable

endorsing the machine-or-transformation test as the sole test for patentability, a majority endorsed this as a “useful and important clue”.¹⁹⁰ The notion that the Information Age requires the modification of the patent regime, that “new technologies may call for new enquiries”¹⁹¹ only attracted minority support.

In Australia, the rush toward a broad view of patentability has also peaked. It was through the reinstatement of a physicality notion that limits were applied to subject matter, through a modification of the useful result test to a “a concrete, tangible, physical, or observable effect [test]”.¹⁹²

In the EU, a physicality requirement is inherent in the notion of technical character or effect. This reflects the influence of German patent law, which had coalesced around a notion of technicality since the Red Dove case in 1969.¹⁹³ The most far-reaching statement of this requirement can be found in the statement of the the German Federal Patent Court that “the concept of technology (Technik) constitutes the only usable criterion for delimiting inventions against other kinds of intellectual achievements, and therefore technicity is a precondition for patentability”.¹⁹⁴ Physicality in the specific sense in which it is meant here then means “solving problems by utilising natural forces”.¹⁹⁵ The natural forces involved are gravity, electromagnetic force, strong nuclear force and weak nuclear force.¹⁹⁶

It is acknowledged however, that within the framework of a physicality requirement, the dominant approach at the EPO favours physicality as a mere form requirement. Any reference to hardware, even a pen and paper¹⁹⁷ is sufficient for patentability in Munich. As noted in Chapter 4 however, clever drafting may get an applicant over the subject matter hurdle, but is unlikely to escape inventive step. UK courts have resisted this approach, and still require a technical (physical) contribution. That this is a physicality requirement is evident from the analysis of the two inventions considered in *Aerotel*. The first, the *Aerotel* invention, was patentable because of the “new physical combination of hardware”¹⁹⁸ featured in the claims. The *Macrossan* application, although claiming a method to be implemented on a computer, did not exhibit the requisite technical character because there was nothing technical (physical) claimed “beyond

¹⁹⁰ *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3258 per Breyer J.

¹⁹¹ *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3228 per Kennedy J.

¹⁹² *Grant v Commissioner of Patents* (2006) 154 FCR 62 at 70.

¹⁹³ BGH GRUR 1696, 672 “Rote Taube”.

¹⁹⁴ BPatG Fehlersuche 2000-07-28: Patentansprüche auf “Computerprogrammprodukt” etc unzulässig, BPatG /17W(pat)69/98, <<http://swpat.ffii.org/analysis/trips/index.en.html>> (11 October 2006).

¹⁹⁵ See Foundation for a Free Information Infrastructure, “Regulation about the invention concept of the European patent system and its interpretation with special regard to programs for computers”, <<http://swpat.ffii.org/stidi/javni/index.en.html>> (12 October 2006). The longhand version of this test, “plan-conformant activity of using controllable natural forces to achieve a causally overseeable success which is, without mediation by human reason, the immediate result of controllable natural forces”, owes its origin to the German Federal Court’s 1977 “Dispositionsprogramm” case. See BGHZiv 1977 Bd 67 p22ff; BGHZ 67, 22; Beschluss des X. Zivilsenats des BGH in der Rechtsbeschwerdesache X ZB 23/74. English translation available at <<http://swpat.ffii.org/vreji/papri/bgh-dispo76/>> (13 April 2007).

¹⁹⁶ For a brief summary see Jeff Silvis and Mark Kowitt, “The Four Forces of Nature,” *Ask An Astrophysicist*, <http://imagine.gsfc.nasa.gov/docs/ask_astro/answers/980127c.html> (9 May 2011).

¹⁹⁷ T258/03 *Hitachi/Auction Method* [2004] EPOR 55 at [4.6].

¹⁹⁸ *Aerotel Ltd. v. Telco Holdings Ltd and in the matter of Macrossan’s Application* [2006] EWCA Civ 1371; [2007] RPC 7 at [53].

the mere fact of the running of a computer program".¹⁹⁹ Why that is insufficiently physical is addressed in detail in Chapter 7.²⁰⁰

As a matter of policy, it is submitted that a physicality requirement, whilst admittedly the most conservative option, serves an important function. On a theoretical level, a physicality requirement is consistent with the theoretical basis for awarding property rights in patent law, namely, that "to create something is to control its existence. Without control there is no possession and no ownership."²⁰¹

On a practical level, the laws of physics and material properties usually offer an inventor a series of difficult technical obstacles to overcome which mean that the inventor runs the risk of complete failure when trying to realise an invention. As opposed to going to market with a bad product, the inventor might not have a product to take to market at all. For example, Dratler notes the following physical limitations to be taken into account in developing a new machine:

While a machine's design may appear operable in concept, in order to work in the real world it must successfully address such practical problems as: metal fatigue, strain, bending, stress fractures, vibration, corrosion, pollution, spalling, differential thermal expansion and contraction, unintended electrolysis, dust, dirt, friction, ablation, evaporation, deterioration of lubricants, electric arcing, unwanted generation of static or other electricity, and aging.²⁰²

Further examples relevant to the design of computer hardware components include issues of heat dissipation, and transistor density.²⁰³ Such hurdles mean that the developer of a new machine faces the problem that "an infinity of possible mechanical configurations meets an infinity of environmental forces; the end result of that interaction is anything but deterministic, anything but predictable".²⁰⁴ Patents thus offer a form of insurance for the holder whilst they address those risks in order to bring their products to market. That is a laudable justification for the award of patent monopolies.

In contrast the developer of an intangible product, such as "pure" software, or a new method of doing business, relies only on a *model* of the real world. In building this model of the real world, it is not "necessary to fully describe and model the entire world".²⁰⁵ Only those characteristics of the real world which are *relevant* to the purpose for which the software is being

¹⁹⁹ *Aerotel Ltd. v. Telco Holdings Ltd and in the matter of Macrossan's Application* [2006] EWCA Civ 1371; [2007] RPC 7 at [72].

²⁰⁰ See Chapter 7, Section on page 275.

²⁰¹ Gregory Stobbs, *Software Patents* (2nd ed, 2000) at 205, discussing the 'common thread' of US patentable subject matter cases based on the expression in *Chakrabarty* that "anything under the sun that is made by man" is patentable.

²⁰² Jay Dratler, Jr., "Does Lord Darcy Yet Live? The Case Against Software and Business-Method Patents," (2003) 43 *Santa Clara Law Review* 823 at 854.

²⁰³ The current trend, known as Moore's law, has seen transistor density double on integrated circuits approximately every 2 years since 1958. The continuation of this trend depends on finding new materials other than silicon from which to build circuits, and will soon reach the physical limits of the current photolithographic manufacturing process. See Wikipedia, "Moore's law" <http://en.wikipedia.org/wiki/Moore's_law> (6 November 2008).

²⁰⁴ Gary Dukarich, "Patentability of Dedicated information Processors and Infringement Protection of Inventions that Use Them" (1989) 29 *Jurimetrics Journal* 135 at 147. As such, Dukarich calls such technologies "non-deterministic".

²⁰⁵ Gary Dukarich, "Patentability of Dedicated Information Processors and Infringement Protection of Inventions that Use Them" (1989) 29 *Jurimetrics Journal* 135 at 141.

Chapter 6: Why Mathematics Is Not Patentable

written are included. As such, the software developer faces only a finite number of variables (although the number of variables may in fact be quite large). “The elimination of irrelevant modelling characteristics and the inclusion of certain relevant ones constitutes the process of abstraction”.²⁰⁶ Because only a limited number of variables are involved,

because every last one can be totalled up and its effects catalogued, then it follow[s] that only a limited number of possible variable interactions exist. All the possible outcomes, all the possible states an information processing machine or software program can enter, myriad though they be, are still predictable and determinable in advance.²⁰⁷

The result of this predictability is a lower degree of risk for the inventor, as acknowledged by Stevens J in *Bilski v Kappos*:

Business innovation ... generally does not entail the same kinds of risk as does more traditional, technological innovation. It generally does not require the same “enormous costs in terms of time, research and development” *Bicron* 416 U.S. @ 480 and thus does not require the same kind of “compensation to [innovators] for their labor, toil and expense” *Seymour v Osborne* 11 Wall. 516, 533-544 (1871).²⁰⁸

The practical impact of these different risk levels are also reflected in the nature of competition in the relevant market, in the way they impact on lead time:

Hardware products produce lead time because reverse-engineering hardware entails a substantial expenditure of time. Would-be imitators must first determine how the new product works and what manufacturing processes were used to produce it, then adapt or build the requisite manufacturing facilities, and then build and test their own products. Software is different. There are typically no manufacturing processes to analyze, and no special factories to set up. Software is written and tested; it is then published, like books, records, or videotapes. It is possible to copy a computer program in seconds and readily reproduce that copy by the hundreds or thousands. It is more difficult, but nonetheless relatively easy, to adapt, translate, or “port” a program, and thereby appropriate much of the value inherent in the original author’s creation. Software, by its nature, lends itself to quick and unexpected duplication and even translation.²⁰⁹

²⁰⁶ Gary Dukarich, “Patentability of Dedicated Information Processors and Infringement Protection of Inventions that Use Them” (1989) 29 *Jurimetrics Journal* 135 at 142.

²⁰⁷ Gary Dukarich, “Patentability of Dedicated Information Processors and Infringement Protection of Inventions that Use Them” (1989) 29 *Jurimetrics Journal* 135 at 142.

²⁰⁸ *Bilski v Kappos* 130 S. Ct. 3218 at 3254 (2010), citing in support: Dan Burk and Mark A Lemley “Policy Levers in Patent Law” (2003) 89 *Vanderbilt Law Review* 1575 at 1618; Michael A Carrier, “Unraveling the Patent-Antitrust Paradox” (2002) 150 *University of Pennsylvania Law Review* 761 at 826; David S Olson, “Taking the Utilitarian Basis for Patent Law Seriously: The Case For Restricting Patentable Subject Matter” (2009) 82 *Temple Law Review* 181 at 231.

²⁰⁹ Anthony L Clapes, Patrick Lynch and Mark R Steinberg, “Silicon Epics and Binary Bards: Determining the Proper Scope of Copyright Protection for Computer Programs” (1987) *UCLA Law Review* 1493 at 1509.

What this highlights is the fast-paced nature of innovation in non-physical endeavours, which is hard to reconcile with the length of protection afforded by the patent grant. It also calls into question the suitability of a regime in which award of monopoly rights, at the end of a sometimes lengthy period of examination, may occur well after the shelf-life of the innovation has expired.

Physicality of the useful arts against intangibility of the fine arts Durham raises an interesting objection to the usefulness of a physicality requirement in distinguishing between the useful and fine arts. Durham notes that

most abstract of “arts” have their physical manifestations and effects on the material world. The field of law, for example, produces contracts and statutes written on paper, and it alters conditions and conduct in the “real world”.²¹⁰

This is true, but it is submitted that this is not a sufficient reason to abandon a physicality requirement as Durham contends. As a matter of intuition, and as a matter of analysis, it is possible to distinguish between the cultural “arts” and technological “making and using”. The key word in Durham’s complaint is “manifestation” — a legal agreement which manifests itself in a written contract is not patentable, because the advance over what existed before is in the agreement, a meeting of the minds, which is clearly non-physical. The issue that Durham draws attention to is one of proper characterisation of an invention. Clever claim drafting can only be properly addressed by looking to the substance rather than the form of claims.²¹¹

Another answer to Durham’s criticism is found in Kant’s classification of the fine arts discussed above.²¹² It was noted that the categories of the fine arts can be classified by the way in which humans communicate. When the fine arts are seen as communications between artist and audience, then it becomes clear why the physicality of the objects produced is not the essence of the fine arts. It is the *message* communicated by the artistic work which is its essence, and which is significant. It is that message which will be evaluated in judging the worth of a particular object independent of any physical functionality - such physical functionality is secondary in importance in the fine arts. With useful arts, the significance lies in “what is achieved in physical fact”,²¹³ the particular physical advantage which the invention offers, which will be adjudged in determining its worth.

Mitcham, whilst acknowledging that both the fine arts and useful arts involve some sort of design process to achieve an end product (which may or may not be physical), sets out in detail the difference between the two processes:

one can ... distinguish engineering design from artistic design according to ideals or ends in view. The engineering design ideal of efficiency stands in contrast to

²¹⁰ Alan L Durham, “‘Useful Arts’ in the Information Age” (1999) *Brigham Young University Law Review* 1419 at 1448.

²¹¹ To this end, the 4-step approach of the Court of Appeal in *Aerotel* provides useful guidance. One reason why the Courts in the UK seem better at characterisation is because it becomes of particular importance in dealing with enumerated exclusions to patentability.

²¹² See page 258 above.

²¹³ *RCA Photophone Ltd v Gaumont-British Picture Corporation* (1936) 53 RPC 167 at 191.

Chapter 6: Why Mathematics Is Not Patentable

the artistic design ideal of beauty. Beauty is not so much a question of materials and energy as of form. About this the whole subject of aesthetics has more to say, whereas it is ethics or politics that would incorporate a philosophical evaluation of efficiency.

Yet the difference between these two types of design does not remain at the level of ideals; it penetrates to the design activity itself. Efficiency refers to a process – is a criterion for choosing between processes or products conceived as functioning units – whereas beauty is in the primary instance a property of stable objects. ... The ends of artistic design must be formal, whereas the final causes of engineering actions are justified in terms of human needs, wants or desires.

A further observation: Engineering design limits itself to material reality (metaphysically, matter and energy are both matter as contrasted with form). This limitation is to be grasped or approached however, by means of a mathematical calculus of forces closely associated with classical physics (Galileo and Newton) and its specific mathematical abstraction. The picturing or imagining that goes on in engineering design is done, as it were, through a grid derived from this physics – the grid itself being articulated, in the first instance, as the engineering science of mechanics. This viewing of matter and energy through the grid of classical physics gives engineering design a rational character not found in art. Engineering images, unlike other images are subject to mathematical analysis and judgment; this is their unique character and one that sometimes leads people to confuse them with thinking in a deeper sense.

Art is also concerned with imagining, but its images cannot be qualitatively analyzed – they are not subject to any well-developed calculus. Thus art, in contrast to engineering, appears as both more intuitive and more dependent on the senses.²¹⁴

This account clearly accords with the analytical framework offered above. It makes clear that engineering (or the useful arts) is both directed towards physical outcomes, and limited during the design process by the “grid of classical physics”, from which Mitcham asserts that the rationality of engineering design emerges, in contrast to the intuitive and sense-dependent design in the fine arts (considered in this chapter so far under the banner of the aesthetic). It is also clear from the quoted passage that the purpose of technology, sourced as it is in physical “human needs, wants or desires”²¹⁵ is distinct from the “formal” (or expressive) ends of artistic design.

The three dimensions of the fine arts versus useful arts distinction

The discussion above leads to a distinction between the fine arts on one hand, and the useful (or technological) arts on the other according to three distinct dimensions of analysis. These are set out in table 6.1.

²¹⁴ Carl Mitcham, *Thinking Through Technology: The Path Between Engineering and Philosophy* (1994) at 229-230.

²¹⁵ Carl Mitcham, *Thinking Through Technology: The Path Between Engineering and Philosophy* (1994) at 229.

Freedom, the useful arts and non-patentability

Fine Arts	Useful Arts
Aesthetic	Rational
Expressive	Purposive
Intangible	Physical

Table 6.1: Fine arts versus useful arts

Firstly, it is asserted that the fine arts are concerned with the aesthetic whereas the useful arts are concerned with the rational. That is, they may be created in order to provoke an aesthetic reaction, or perhaps more importantly in the current context, aesthetic considerations motivate their practitioners. In contrast, technology is rational, being centred on the process of design. Its practitioners are driven by the desire to achieve efficiency gains. Secondly, it is contended that fine arts are expressive whereas the useful arts are purposive. They are sometimes said to be ‘ends in themselves’, which is to say that the final form in which they manifest themselves is secondary in importance to the expression that they represent. Technology on the other hand is purposive, being designed to solve a particular problem, to achieve a particular result. Thirdly, it is suggested the preferable interpretation of the useful arts is that they deal with the physical. Fine arts on the other hand are said to be intangible in nature. Whilst they may (and indeed often are) manifested in a physical form, their essence, or importance, lies beyond that manifestation.

Each of these dimensions are now considered in turn.

Aesthetic versus rational

It has been demonstrated that mathematics is primarily a mental activity.²¹⁶ Therefore it falls to determine whether it is a mental activity which is aesthetic, in that it is based on intuitive, sense-dependent considerations, or one which is rational, in that it is a rational, engineering-like activity which is directed towards the satisfaction of the physical wants, needs and desires of humanity.

The role of logic and the apparent availability of objective criteria upon which to judge a mathematical proof weighs in favour of a view of mathematics as rational activity. Yet this objectivity is to an extent superficial and misleading. The unresolved foundational crisis in mathematics, discussed in Chapter 5, makes it clear that whilst mathematicians seek to remove any reliance on intuition, this cannot be done. Irrational numbers, imaginary numbers, and non-Euclidean geometries were all rejected as impossible and irrelevant at various stages through history, yet their usefulness to (often unexpected) real-world applications saw their eventual validity. Conjectures and hypotheses, whilst unproven, still act as a very real motivation to mathematicians, and it has been through searching for proofs that interesting mathematics has come about.

Perhaps the most telling statement of the limits of objectivity and logic in mathematics was Gödel’s incompleteness theorem, by which he demonstrated the impossibility of excluding

²¹⁶ See Section 4 above.

Chapter 6: Why Mathematics Is Not Patentable

uncertainty, in the form of incompleteness or inconsistency, from any powerful mathematical system.

Further, the role of the aesthetic in motivating mathematicians, and generating mathematical advances was explored earlier in the chapter. Thus it seems appropriate to conclude that the balance of evidence supports the primacy of the aesthetic over the rational in the field of mathematics.

Expressive versus purposive

On any account of mathematics, it must still be acknowledged that mathematics is useful, in that it is often a tool which is used by scientists and engineers in their work. However, the role of expression in mathematics has also been documented. Mathematics is expressed in a symbolic language, which is unlike “natural” language, but which is nonetheless expressive.

The traditional status of mathematics as one of the liberal arts curriculum also confirms this categorisation. It is obvious from the content of the enumerated liberal arts that mathematics was not only a substantial component, but also that mathematics was considered a form of higher learning, based as it was in the Quadrivium, rather than the Trivium. Therefore, on any understanding of the term liberal arts, mathematics is clearly within its ambit.

Intangible versus physical

The role of mathematics in explaining real-world phenomena in the physical sciences might be thought to lend weight to the classification of mathematics as directed towards the physical. It may have some relevance to human needs, wants and desires to the extent that mathematical advances lead to better understanding of the physical world, and thereby to better utilisation of it. However, this is at best an indirect physical relevance.

It was established above that mathematics is primarily a mental activity, and whilst it is often expressed in a symbolic language, that physical manifestation is not the essence of mathematics, or the grid through which its development might be viewed. To the extent that mathematics is manifested in physical form, its significance lies not in the production of printed documents. Even on the formalist account of mathematics being nothing more than a game played with symbols on paper, it is clear that mathematics is the game, not the symbols.

Summary

The lesson to be learned from this understanding of mathematics, and the importance of freedom to innovation in the field, is that it is a trap to focus on the “usefulness” of mathematics, at the expense of a critical analysis of the nature of mathematical creativity – its intangible, expressive and aesthetic characteristics. If there is to be a way forward, it must be to balance the benefit to society of encouraging inventive or creative activity against the impact of awarding monopoly rights on the interests of others practising in the field. This sort of utilitarian analysis

should lie at the heart of the patentable subject matter inquiry. As touched on both in Chapter 1 and in the discussion of the formalist school in both Chapter 5 and this chapter, mathematics and computer software are isomorphic, so it follows that software would be expected to be non-patentable as well.

6 Conclusion

This chapter has explored the role of the freedoms of thought and expression in mathematics. Through that exploration, the nature of mathematics as a primarily mental activity, expressed in a symbolic language, and driven forward by aesthetics has become apparent. It was then demonstrated how the need for such freedoms can be reconciled with an understanding of mathematics as being a fine art rather than a useful art, and hence outside the scope of patentable subject matter. In establishing the distinction between fine and useful arts, a 3-dimensional analysis was adopted, and it was demonstrated how on each dimension of the analysis, mathematics falls within the fine arts.

Having thus established the role of freedom within the mathematics exception, and its proper classification outside the scope of patent law, There is an immediate flow-on consequence to the patentability of software in two respects. Firstly, the isomorphism between mathematics and software development means that, at least as a starting point, it should be expected that software is not patentable subject matter, being a similarly creative activity. Secondly, it is now possible to bring a greater deal of clarity to the related field of computer software, using that understanding of the non-patentability of mathematics, and the analytical tools developed to consider if or when software should be considered patentable. It is to that task that the next chapter of this thesis is devoted.

Why Programming is not among the Useful Arts

*It seems beyond question that the machines – the computers – are in the technological field, are part of one of our best-known technologies, and are in the “useful arts” rather than the “liberal arts,” as are all other types of “business machines,” regardless of the uses to which their users may put them. How can it be said that a process having no practical value other than enhancing the internal operation of those machines is not likewise in the technological or useful arts?*¹

1 Introduction

The last chapter demonstrated that the requirements of mathematical activity locate it properly outside the useful arts. It will be recalled from Chapter 1 that there is an isomorphism, or structural similarity, between the activities involved in mathematics and programming.² In the discussion of formalism in Chapter 5, it was briefly noted that there is a link between formalism and the development of a formal theory of computing.³ This chapter will consider how far the isomorphism and formal connection extend the asserted basis of the non-patentability of mathematics to software.

As the previous chapter did for mathematics, the expressive, aesthetic and intangible nature of software development will be explored. The existence of these attributes, it will be argued, supports the contention that the classification of mathematics as a fine art will usually be appropriate to software. In particular, it will be suggested that the interrelationship of software and hardware, referred to as context in Chapter 1, acts as an important consideration, which becomes more important the higher up the abstraction chain one goes. Where the software and hardware are closely interrelated, the physical limitations may constrain the software such that

¹ *In re Benson* 441 F.2d 682 (CCPA, 1971) at 688.

² See Chapter 1, at 4.

³ See Chapter 5, at 1.

Chapter 7: Why Programming is not among the Useful Arts

expressive and aesthetic considerations are similarly limited. Where software is merely ancillary to a physical device however, this does not change the nature of programming, but it does in effect change what is sought to be patented. The limited scope of the claims will mean both that programming as a creative activity is not disrupted, and that what is claimed is sufficiently connected to the traditional conception of the useful arts that is not contentious. Determining whether it is software, or a physical device which is actually being claimed requires difficult determinations of fact in any particular case, but the difficulty of the choice does not obviate the need to draw such distinctions.

2 Is programming a fine art or a useful art?

In Chapter 1, the structural similarities between software development and mathematics were introduced.⁴ It may be recalled that according to the Curry-Howard isomorphism, the activities involved in doing mathematics map perfectly onto the activities involved in developing software. This being the case, the fact that mathematics is a fine art because it is aesthetic rather than rational, expressive rather than functional, and abstract rather than physical suggests that software deserves a similar classification. But the relationship between computer hardware and software is clearly a complicating factor. The question is thus whether the relationship thereby changes the nature of software development, or programming, such that it is properly considered a useful art.

To explore the patentability of software, the 3-dimensional analytical framework applied to mathematics in the last chapter will be followed. That analysis requires a consideration of:

- whether software is an intangible or physical entity;
- whether software is an expressive or purposive artefact; and
- whether the creation of software is dominated by an aesthetic or rational approach.

Computer hardware, as the physical machine on which software executes, *prima facie*, has a direct impact on the physicality/intangibility dimension.⁵ Therefore that dimension is considered first. However it is conceivable that it may have an effect, directly or indirectly, on the other two dimensions of analysis. After consideration of each individual dimension, the nature of software development as a whole will be considered, before a concluded view is reached about the nature of software.

Before commencing that analysis, it is apposite to note that in this chapter, it is programming, rather than software, which is analysed. There are two reasons for this. The first emerges from the analysis of mathematics undertaken in Chapter 5. It may be recalled that there was an

⁴ See page 33 and following.

⁵ That the execution of software on a general purpose computer creates a “new machine” tailored to a specific purpose “has some merit as a matter of computer science.”: Pamela Samuelson, “Benson Revisited: The Case Against Patent Protection for Algorithms and Other Computer Program-Related Inventions,” (1990) 39 *Emory Law Journal* 1025 at 1045, note 63. See the discussion of the problems with this approach in Chapter 4, at 164. The problems with a purely physical analysis of computer software are also discussed in section 2 below.

irreconcilable difficulty attending on a similar inquiry into the nature of mathematics. Various schools of philosophy had posited theories, none of which became dominant, and many of which were in direct conflict with each other. This led to a decision to change the focus of the inquiry from what mathematics *is*, to what mathematicians *require*. It was through that change of focus that a useful distinction between patentable and non-patentable subject matter was derived in Chapter 6. Similarly, the focus in this chapter will not dwell on what software *is*, but on what programmers *require* to advance the art.

Looking at it another way, it is the activity of programming which is important, because it is that activity which is regulated by the grant of patent. Starting with a concept, and ending with executable code, the programmer determines the process or algorithm with precision, describing in detail the computable steps of the process.

Intangible or physical?

At the origins of modern computer, the “programmer’s” task involved a physical reconfiguration of the hardware, plugging in wires and throwing switches. Such “programs” were “clearly physical and as much a part of the computer system as any other part.”⁶ Although the days of physical engagement with the hardware are gone, even in their modern form, computer programs can be understood physically, either as transforming a general purpose computer into a “new machine” running the specific program, manifested in the form of electrons pulsing through the circuitry of the computer.⁷ Alternatively, they may be understood as a series of numbers stored in magnetic or other form on a carrier.⁸ Indeed these characterisations are exactly those relied on by various courts, as discussed in Chapter 4.

However, computers generally, and computer software specifically, can be understood both at a physical level and at a symbolic level.⁹ Indeed, “it is the understanding on the symbolic level which makes computers calculating devices, for it is under this kind of interpretation that various structures or processes of the computer are understood as symbols.”¹⁰ It is both pointless, and almost physically impossible, to attempt to describe the operation of a modern-day software package, such as Microsoft Word, by describing it by reference to the motion of electrons around the circuitry of the computer hardware. Even moving a level above to the realm of machine code, to attempt to describe Microsoft Word by detailing the logical operations performed by the CPU on bits stored in memory is futile. Even describing one particular function, such as counting the number of words in the currently open document, at this level is both difficult to comprehend, and failing to see the forest for the trees. Moving any further beyond this is to stray away from the purely physical into the realm of the symbolic.

Therefore a strictly physical analysis of software ignores the very essence of software. Brooks, in a seminal article addressing the failure of software to mark productivity advances in hard-

⁶ See James H Moor, “Three Myths of Computer Science” (1978) 29(3) *British J Phil Sci* 213 at 213.

⁷ A discussion of this characterisation appears in Chapter 4, at page 2.

⁸ See Chapter 4, starting at page 168.

⁹ James H Moor, “Three Myths of Computer Science” (1978) 29(3) *British J Phil Sci* 213 at 213, 215.

¹⁰ James H Moor, “Three Myths of Computer Science” (1978) 29 *British J Phil Sci* 213 at 213.

Chapter 7: Why Programming is not among the Useful Arts

ware,¹¹ captures the essence of software as “a construct of interlocking concepts; data sets, relationships among data items, algorithms and invocations of functions.”¹² Generally speaking, these “objects of computer science are abstractions subject to logical — but not physical — constraints.”¹³

A misunderstanding of this dual nature of software is also evident in case law in both the US and Australia which has used the term algorithm as if it is interchangeable with the term software.¹⁴ Algorithms are better understood as either a marker for the purely symbolic aspects of a program,¹⁵ the “mathematical counterpart of a textual object that is the program”,¹⁶ or as standing at a level of abstraction above programs, to the extent that programs are one of a class of possible implementations of a particular algorithm.¹⁷ On any of these interpretations, the connection with the physical level is either absent, or significantly weakened.

The issue then is whether the symbolic or physical nature of software is of greater weight in the present context. The short history of software discussed in Chapter 1 demonstrates that the trend has been towards greater and greater abstractness, through

the creation of tools, techniques, and computing hardware which permit programmers to be increasingly ignorant of the material realities of the machine, focusing instead on the abstractions they create and manipulate.¹⁸

It might seem like sophistry to suggest that a process, software, executed on a physical device, a computer, and on which its execution can be observed, could nevertheless be characterised as non-physical. But this is not what is suggested. It is indeed possible to understand software at a purely physical level. Neither is it to be denied that there is a relationship between the creation of software, and its execution.¹⁹ But software cannot be understood only at that physical level.

¹¹ Brooks notes that despite “desperate cries for a silver bullet – something to make software costs drop as rapidly as computer hardware costs do ... [t]here is no single development, in either technology or in management technique, that by itself promises even one order-of-magnitude improvement in productivity, in reliability, in simplicity.”: Frederick P Brooks Jr, “No Silver Bullet: Essence and Accidents of Software Engineering” (1987) 20(4) *Computer* 10 at 10.

¹² Frederick P Brooks Jr, “No Silver Bullet: Essence and Accidents of Software Engineering” (1987) 20(4) *Computer* 10 at 11.

¹³ Timothy Colburn and Gary Shute, “Abstraction, Law and Freedom in Computer Science” (2010) 41(3) *Metaphilosophy* 345 at 346. See also

¹⁴ The origins of this approach lie in *Gottschalk v Benson et al* (1973) 409 U.S. 63, discussed in Chapter 3 at 3 on page 86.

¹⁵ “In the absence of a precise semantics, ... programs are just meaningless scribbles; to read them as algorithms, we must first interpret the language – and it is the *meanings* attached to programs by this interpretation which are the algorithms, not the programs themselves.”: Yiannis N Moshovakis, “On founding the theory of algorithms”, in Harold G Dales and Gianluigi (eds), *Truth in Mathematics* (Oxford University Press, 1998) 71 at 79.

¹⁶ Raymond Turner and Ammon Eden, “The Philosophy of Computer Science” *Stanford Encyclopedia of Philosophy* (Summer 2009 Edition) <<http://plato.stanford.edu/archives/sum2009/entries/computer-science/>> (25 July 2011).

¹⁷ See Yiannis N Moshovakis, “On founding the theory of algorithms”, in Harold G Dales and Gianluigi (eds), *Truth in Mathematics* (Oxford University Press, 1998) 71 at 75.

¹⁸ Scott Dexter et al, “On the Embodied Aesthetics of Code” (2011) 12 *Culture Machine* 1 at 11.

¹⁹ The exact nature of that relationship has not been settled. Whilst it is tempting to say that the text of the source code and executable machine code have a causal relationship, in that it is through the process of compilation that one causes the other. However this ignores what is often a complex, context-specific chain of events, that may in fact be of particular significance in determining the nature of what is claimed. For example, a program written in a language, might be capable of compilation via a number of different implementations of a language, each with their own set of dependencies lower down the abstraction stack (see the discussion of the Python language

Is programming a fine art or a useful art?

The more significant understanding of software for present purposes is on the symbolic level. To understand why, it is again useful recall the difficulties attendant on determining the nature of mathematical objects. As noted above, it is suggested that with the isomorphic activity of software development, what is *required* to advance the art is a more useful avenue of inquiry.²⁰

This being the case, to properly understand how a concern for the intangible emerges from the physical execution of software, and indeed comes to eclipse it, it is necessary to briefly diverge back into the history of mathematics. This is because it was in mathematics that the process by which mental processes might be externalised was developed. The emergence of the mechanical device by which such processes might be performed without human intervention therefore depends both conceptually and historically on these efforts.

The externalisation of thought

The ability to externalise, or mechanise the mental processes owes its origins to the work of mathematicians such as Gottfried Leibniz and George Boole to “capture patterns of thought by means of algebra”.²¹ An (arguably) more modest goal was the later attempt by Bertrand Russell and Alfred North Whitehead to reduce all of mathematics to symbolic logic, in their work *Principia Mathematica*. That attempt was demonstrated to be a failure by Kurt Gödel. However, within his proof, Gödel used an ingenious device which deserves further attention.²² Gödel recognised that any formal language could be described as a string of unique symbols. For example,²³ the following statement:

$$0 < S0$$

Contains the symbols “0”, “<” and “S”. Gödel’s trick involves substituting for each well-formed statement, a unique number.²⁴ Valid deductions within the mathematical system can then be “mapped” on to operations on those numbers, thereby reducing symbolic logic to mathematical calculation. It is this ability to understand something at both a symbolic level, and at the

implementation at 288 below). Or it may be that the source code is to be used as a blueprint for the realisation of the process in hardware. See the discussion in Raymond Turner and Ammon Eden, “The Philosophy of Computer Science” *Stanford Encyclopedia of Philosophy* (Summer 2009 Edition)

<<http://plato.stanford.edu/archives/sum2009/entries/computer-science/>> (25 July 2011) at section 2.1.

²⁰ It might also be noted that there is a correlation between the philosophy of mathematics and a philosophy of software here. To equate software with its physical instantiation is to take an empiricist view of software. A quasi-empiricist might however suggest that software is not *identical* with its physical instantiation, but is validated by it, with the symbolic aspects of software forming a web of belief around it. On the various internalist views, software is the abstract algorithms and data structures, which exist either only as products of the human mind, or externally in the Platonist realm. On these views, source code is merely a description of those objects, which as a side-benefit can be manifested in an executable form on a computer, if described precisely enough.

²¹ Keith Devlin, “The mathematics of human thought” in *Devlin’s Angle*, Mathematical Association of America, January 2004 <http://www.maa.org/devlin/devlin_01_04.html> (22 July 2011).

²² Gödel’s proof is necessarily complicated. What follows is not a complete explication of the proof, but a summary of the techniques used which will be shown to be relevant to software development.

²³ This example is taken from Wikipedia, “Proof sketch for Gödel’s first incompleteness theorem” <http://en.wikipedia.org/wiki/Proof_sketch_for_G%C3%B6del's_first_incompleteness_theorem> (22 July 2011), which is in turn derived from the version used by Hofstadter in Douglas R Hofstadter, *Gödel, Escher, Bach: An Eternal Golden Braid* (Basic Books, 1980).

²⁴ For example, if we assign to those symbols above, the numbers “666”, “212” and “123”, the statement above can be represented as 666 212 123 666.

Chapter 7: Why Programming is not among the Useful Arts

numerical level, which lies at the heart of computing. The computing hardware, operating as a mathematical calculation device, can perform processes which are also capable of being understood at the symbolic level by humans.

The difference with computers is that an additional layer of understanding exists below the numerical, wherein physical switches are represented as numbers, by assigning '1' to the on position, and '0' to the off position.

When characterised in this way, it can be seen how the modern computer is an extension of the logicist programme. Hilbert's work on the *Entscheidungsproblem* inspired both Gödel's work, his incompleteness theorem containing one of the first definitions of "general recursion", and also Turing's definition of "effective computability".²⁵ Although the links between Turing and Gödel are only circumstantial,²⁶ Turing's model of computation shows its historical origins in mathematics. This is nowhere more apparent than in Turing's comparison of "a man in the process of computing a ... number to a machine."²⁷ That is, the original Turing machine was intended by Turing to model the process of *human* computation. Put another way:

[B]y carrying out computations according to a selected plan, the mathematician acts in a way similar to a Turing machine: in considering some position in his writings and being in a certain "state of mind", he makes the necessary alterations in his writing, is inspired by a new "state of mind", and goes on to contemplate further writing. The fact that he completes more complicated steps than a Turing machine seems not principally significant.²⁸

It will be recalled that this was exactly the basis on which the mental steps cases sought to exclude patentability, namely that a computerised process could also be carried out by a human operator. But the problem with the mental steps doctrine is that it tends to equate thought and computation:

The ... mantra that cognition is computation, promulgated by early cognitive science researchers, defined thought in exactly those terms that could be instantiated

²⁵ "Historically, proposals were made by a number of different persons at about the same time (1936), mostly independent of one another, to identify the concept of effectively calculable function with various precise concepts. ... These notions, which ... were quite different in formulation, have all been proved equivalent in the sense that the classes of functions obtained are the [same]": Martin Davis, *Computability and unsolvability* (Courier Dover Publications, 1982) at 11-12.

²⁶ There is some evidence that Turing's work on computability was inspired by the work of both Gödel and Hilbert (see Edward R Griffor, *Handbook of computability theory* (Elsevier, 1999) at 11). There is also evidence of Gödel's direct influence on, but subsequent disapproval of, Church's work (see for example Griffor at 8-11). Finally, there is evidence that Gödel "enthusiastically accepted Turing's thesis and his analysis, and thereafter always gave credit to Turing (not to Church or to himself) for the definition of mechanical computability and computable function.": Griffor at 11.

²⁷ Alan M Turing, "On Computable Numbers, with an Application to the Entscheidungsproblem" (1936-1937) 42(2) *Proceedings of the London Mathematical Society* 230 at 231.

²⁸ SpringerLink, "Turing machines", in *Encyclopaedia of Mathematics* (2001) <<http://eom.springer.de/t/t094460.htm>> (22 July 2011).

Is programming a fine art or a useful art?

by the digital computer, downplaying significant other aspects of the human mind, such as motivation, emotion and cross-cultural differences...²⁹

In fact, Gödel's incompleteness theorem is a demonstration of the falsity of equating cognition and computation. His work demonstrated the difference between our intuitive understanding of concepts, and attempts to express them. This fuzziness is also evident in Turing's conception of what is computable. Thus it is asserted that the work of the programmer is in large part concerned with bridging this gap between the human conception of a process, and the precise description of a computable process in syntactically correct, executable machine form – the semantic gap.³⁰ Therefore, although it is true to say that computable processes can be understood at the physical level, and equivalently, that software can be understood as executing on computer hardware, the work of the programmer in most cases takes place a long way removed from these physicalities, and it is only in limited circumstances where software will be limited in any real sense by physicality.

The speed by which electronic computation is carried out makes possible more complex computations than would ever be practically reasonable for a human to carry out.³¹ But this should not mean that the relationship between software and mental processes should be overlooked. Those who focus on the speed of the calculation itself overlook the sustained effort of the programmer to translate their own mental processes into a series of mechanical steps which can then be performed by a computer. Perlis captures that progression from mind to machine as follows:

Every computer program is a model, hatched in the mind, of a real or mental process. These processes, arising from human experience and thought, are huge in number, intricate in detail, and at any time only partially understood. They are modelled to our permanent satisfaction rarely by our computer programs.³²

The art of abstraction is a tool which is directed to closing this gap as much as possible. Just how much abstraction is appropriate involves a trade-off:

²⁹ Scott Dexter et al, "On the Embodied Aesthetics of Code" (2011) 12 *Culture Machine* 1 at 10. See also Peter Naur, "Computing versus human thinking" (2007) 50(1) *Communications of the ACM* 85 - in which Naur puts forward an interesting conception of computing as a form of description. These other aspects of the human mind are explored in the other two dimensions of the analysis, below. Davis and Hersh also discuss the limits of the "mathematization" of the world in Philip J Davis and Reuben Hersh *Descartes' Dream; The World According to Mathematics* (Dover Publications, 1986).

³⁰ For an overview, see Wikipedia, "Semantic gap" <http://en.wikipedia.org/Semantic_gap> (25 July 2011). The semantic gap is particularly evident in natural language processing and image retrieval. On the latter see Arnold WM Smeulders et al, "Content-Based Image Retrieval at the end of the Early Years" (2000) 22(12) *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1 at 1: "There is something about Munch's 'The Scream' or Constable's 'Wivenhoe Park' that no words can convey. It has to be seen. The same holds for a picture of the Kalahari desert, a dividing cell, or the facial expression of an actor playing King Lear. It is beyond words. ... Pictures have to be seen and searched as pictures: by object, by style, by purpose."

³¹ "Electronic computers are intended to carry out any definite rule of thumb process which could have been done by a human operator working in a disciplined but unintelligent manner. The electronic computer should however obtain its results very much more quickly." Alan Turing, "Programmers' Handbook for Manchester Electronic Computer" *University of Manchester Computing Laboratory*, <http://www.AlanTuring.net/programmers_handbook> (21 July 2008).

³² Alan J Perlis, "Foreword," in Harold Abelson and Gerald Jay Sussman, *Structure and Interpretation of Computer Programs* (2nd ed, MIT Press, 1996) at xi.

Chapter 7: Why Programming is not among the Useful Arts

You can ‘ignore’ the human concerns and end up with machine code ... You can ignore the machine and come up with a beautiful abstraction that can do anything at extraordinary cost and/or lack of intellectual rigor.³³

Meaningful limitations It is clear then that the key issue will be one of determining the extent to which a hardware device constrains the development of software in a meaningful sense. Thus the determination of whether claims over software are physical or intangible will involve difficult questions of degree. However, claims to a general purpose computer, or “any hardware” are unlikely, of themselves, to warrant a conclusion that some form of *meaningful* limitation is involved. It is only likely to be in limited circumstances where the physical aspects of a computer dictate are central to the design and implementation of the software which runs on it. It is suggested that size, performance or a need to communicate directly with specific hardware is when physical limitations come into play. However Moore’s law and practical experience suggest that it will be rare that the resource constraints exert any real limiting influence. Goodliffe suggests that the following contexts may require that the physical resources of the computer be a central concern:³⁴

- game programming
- digital signal processing
- resource-constrained environments (for example deeply embedded systems)
- real-time systems³⁵
- numerical programming³⁶

But there is a need to distinguish optimization from physicality. The desirability of having code which efficiently executes on a computer is not necessarily the same as the physical limitations imposed on traditional inventions. This is because the physical limitation may be arbitrary, in the sense that optimisation is but one alternative,³⁷ and one which is best avoided, about 97% of the time.³⁸ Further, greater efficiency may be achieved by the use of a better design, or a better algorithm – that is, a non-physical solution can be employed.³⁹

³³ Bjarne Stroustrup in Federico Biancuzzi and Shane Warden, *Masterminds of Programming* (O’Reilly Media, 2009), at 5.

³⁴ Pete Goodliffe, *Code craft: the practice of writing excellent code* (No Starch Press, 2007) at 205.

³⁵ For example, in flight navigation systems or medical equipment.

³⁶ For example some applications in the financial sector and scientific research require the processing of very large data sets.

³⁷ Alternatives to optimisation might include: using faster hardware; optimising the various programs running on the hardware; asynchronously executing slow code; hiding the slow execution behind a responsive user interface; making the process run unattended; or using a new compiler: Pete Goodliffe, *Code craft: the practice of writing excellent code* (No Starch Press, 2007) at 204.

³⁸ “We should forget about small efficiencies, say about 97% of the time: premature optimization is the root of all evil.”: Donald E Knuth, “Structured Programming with go to Statements” (1974) 6(4) *Computing Surveys* 261 at 268. This is because optimisation can impact on other desirable qualities of code, for example, readability, simplicity and maintainability or extensibility. The issue of what is valuable in code is addressed in the section on aesthetic versus rational considerations below. Optimisation might also be a source of conflict, in that what is optimal on one hardware system, is suboptimal on another. Finally, it may simply be that the extra manual labour involved is not warranted. Pete Goodliffe, *Code craft: the practice of writing excellent code* (No Starch Press, 2007) at 203.

³⁹ The numerical programming examples listed above may well often fit into this category.

Is programming a fine art or a useful art?

It is only really then when software must interact with a particular piece of hardware, and where the optimising compiler has no knowledge of the instruction set, that the instructions of the hardware device constitute a physical limitation on the design. The contexts in which this is likely to occur are in development of the lower levels of the operating system, such as kernel development; writing device drivers and bootloaders, or perhaps when writing platform-specific code for compilers or interpreters.⁴⁰

Summary

In most cases, software is likely to be more intangible in nature than physical, although it may be sufficiently physical in a limited range of circumstances.

Expressive or purposive?

Software is often written to achieve a particular purpose. It is common to define a computer program by reference to that function.⁴¹ As noted above, software is big business, and software performs functions useful to all manner of enterprises, from controlling industrial processes to assisting academics to write large documents. Software products can be purchased as off the shelf components sold on the mass market, or written by a friend as a favour. The design, construction and purchase of software would therefore seem to be dominated by a desire for function. Therefore it is tempting to place great weight on functionality. Indeed to the non-programmer, the notion of software as expressive may seem entirely at odds with their experience of it.

In assessing any computer-related invention, it must be remembered that the programming is done in a computer language. The computer language is not a conjuration of some black art, it is simply a highly structured language. Analogously, if a person were to express a complete thought in German, it would be no trick for a translator to convert that thought into a palpable English form. The thought, thus expressed, might not be worthy of Shakespeare, but it would be understandable to one who uses the English language. Similarly, the conversion of a complete thought (as expressed in English and mathematics, i. e., the known input, the desired output, the mathematical expressions needed and the methods of using those expressions) into the language a machine understands is necessarily a mere clerical function to a skilled programmer.⁴²

However with a few examples it will be shown how software can express or communicate an idea. The programming language Perl is fertile ground for exploring this notion, since it is constructed in such a way that it is possible to write functioning code which directly engages visual perceptions. The high water mark of such code would have to be the program in Listing 7.1. This is, strictly speaking, a fully functional program, although its function may be of little

⁴⁰ This will not always be the case though, as in many cases, some or all of a compiler might be written in a high level language. See the discussion of the Python language below on page 288.

⁴¹ "A computer program is a set of instructions designed to cause a computer to perform a function or produce a particular result.": *Computer Edge v Apple* (1986) 161 CLR 177 at 178.

⁴² *In re Sherwood* 612 F.2d 809 at fn 6.

Listing 7.1 Camel Code

economic value since it draws four smaller camel pictures, using its own visual layout as a template.⁴³ A similar example, set out in Listing on the current page,⁴⁴ demonstrates that it is possible to marry such visual expressiveness with useful functionality.

This decryption program, known as DeCSS, deciphers an anti-copying measure embedded in the DVD format, known as region coding. The code above is a reformulation of the original DeCSS package, which was the subject of both criminal and civil litigation.⁴⁵ The algorithm has been expressed in a number of alternative forms, from mathematical descriptions,⁴⁶ to

282

Listing 7.2 DeCSS Code

```
#define m(i)(x[i]^s[i+84])<<

unsigned char x[5] ,y,s[2048];main(
n){for( read(0,x,5 );read(0,s ,n=2048
); write(1 ,s,n) )if(s
[y=s [13]%8+20] /16%4 ==1 ){int
i=m( 1)17 ^256 +m(0) 8,k =m(2)
0,j= m(4) 17^ m(3) 9^k* 2-k%8
^8,a =0,c =26;for (s[y] --=16;
--c;j *=2)a= a*2^i& 1,i=i /2^j&1
<<24;for(j= 127; ++j<n;c=c>
y)
c

+=v=i^i/8^i>>4^i>>12,
i=i>>8^y<<17,a^=a>>14,y=a^a*8^a<<6,a=a
>>8^y<<9,k=s[j],k
&7)+2^"cr3sfw6v;=k+>/n."[k>>4]*2^k*257/
8,s[j]=k^(k&k*2&34)*6^c+~y
;}}
```

plain English,⁴⁷ to haiku,⁴⁸ and even as a prime number.⁴⁹ All of these reformulations, including the one in Listing ??, were written in response to the uncertainty around whether software, as a form of expression, should be entitled to constitutional protection under the First Amendment.⁵⁰

The cynical reader may see these examples as nothing more than exceptions which make the rule. After all, the US Supreme Court has rightly noted that “it is possible to find some kernel of expression in almost any activity a person undertakes”.⁵¹ Indeed, there is some empirical evidence which suggests programmers see functionality as the dominant consideration.⁵² On this basis it is tempting to suggest that software is “a fundamentally utilitarian construct even assuming it embodies some expressive element”.⁵³

Encryption software has been an important focal point for judicial consideration of the expressivity of software. And in that context, it is now accepted that at least one aspect of software,

⁴⁷ See <<http://www.cs.cmu.edu/~dst/DeCSS/Gallery/plain-english.html>> (15 July 2008).

⁴⁸ See <<http://www.cs.cmu.edu/~dst/DeCSS/Gallery/decss-haiku.txt>> (15 July 2008)

⁴⁹ This 1401 digit number, when represented in binary form is a zipped copy of the original C source code of the descrambler. See <<http://www.utm.edu/research/primes/curios/48565...29443.html>> (15 July 2008).

⁵⁰ The DeCSS campaign was sparked by a preliminary ruling by District Court Judge Kaplan in *MPAA v Reimerdes, Corley and Kazan* (NY, 2 February 2000) which suggested firstly that it was “far from clear” that source code was expressive, and that even assuming some expressivity in source code, “the expressive aspect appears to be minimal when compared to its functional component”. Touretzky, maintainer of the website Gallery of CSS Descramblers, describes his motivation for compiling the alternative DeCSS expressions thus:

If code that can be directly compiled and executed may be suppressed under the DMCA, as Judge Kaplan asserts in his preliminary ruling, but a textual description of the same algorithm may not be suppressed, then where exactly should the line be drawn? This web site was created to explore this issue, and point out the absurdity of Judge Kaplan’s position that source code can be legally differentiated from other forms of written expression.

See John D Touretzky, “Gallery of DeCSS Descramblers” (2000)
<<http://www.cs.cmu.edu/~dst/DeCSS/Gallery/>> (11 July 2011).

⁵¹ *City of Dallas v Stanglin* 490 US 19 (1989) at 25.

⁵² Scott Dexter *et al*, “On the Embodied Aesthetics of Code” (2011) 12 *Culture Machine* 1 at 6, citing A Kozbelt *et al* “Beautiful Software: Characterizing Aesthetic Judgment Criteria of Code Amongst Expert and Novice Computer Programmers”, paper presented to 2010 Biannual Meeting of the International Association of Empirical Aesthetics, Dresden, Germany.

⁵³ *Universal City Studios Inc v Reimerdes* 82 F. Supp.2d 211 (SDNY, 2000) at 226 per Judge Kaplan.

Chapter 7: Why Programming is not among the Useful Arts

namely source code, is expressive. In *Bernstein v United States Department of Justice*,⁵⁴ it was held both at first instance, and on appeal, that source code was expressive and therefore protected by the right of freedom of expression contained in the First Amendment. Source code was “protected speech” because “[t]he distinguishing feature of source code is that it is meant to be read and understood by humans and that it can be used to express an idea or a method.”⁵⁵ Indeed, some programmers see this communicative aspect as the most important consideration in programming.⁵⁶

The use of source code as a way of communicating and representing ideas was also recognised in *Junger v Daley*:⁵⁷

The issue of whether or not the First Amendment protects encryption source code is a difficult one because source code has both an expressive feature and a functional feature. The United States does not dispute that it is possible to use encryption source code to represent and convey information and ideas about cryptography and that encryption source code can be used by programmers and scholars for such informational purposes. Much like a mathematical or scientific formula, one can describe the function and design of encryption software by a prose explanation; however, for individuals fluent in a computer programming language, source code is the most efficient and precise means by which to communicate ideas about cryptography.

...

Because computer source code is an expressive means for the exchange of information and ideas about computer programming, we hold that it is protected by the First Amendment.⁵⁸

Eventually it is translated to a form which is executable on a particular piece of computer hardware. But it maintains its descriptiveness of that computable process. This question then arises: by the time it has taken its executable form, has software has lost its expressiveness? This issue has been dealt with in the copyright context. A suitable example from the Australian jurisdiction is *Computer Edge v Apple*.⁵⁹ At trial it was held that object code was not suitable subject matter for copyright, because it did not fit the definition of a literary work, not being “intended to afford information, instruction or pleasure in the form of literary enjoyment”.⁶⁰

⁵⁴ 176 F.3d 1132 (9th Circuit, 1999).

⁵⁵ *Bernstein v United States Department of Justice* 176 F.3d 1132 (9th Circuit, 1999) at 1140.

⁵⁶ “Programs must be written for people to read, and only incidentally for machines to execute.”: Harold Abelson and Gerald Jay Sussman, *Structure and Interpretation of Computer Programs* (2nd ed, MIT Press, 1996) at xvii. Note that this “readability” is also considered in the discussion of what makes “great code” below.

⁵⁷ 209 F.3d 481 (6th Circuit, 2000).

⁵⁸ 209 F.3d 481 (2000) at 485.

⁵⁹ *Apple Computer Inc v Computer Edge Pty Ltd* (1983) 1 IPR 353, 50 ALR 581 (first instance); *Apple Computer Inc v Computer Edge Pty Ltd* (1984) 1 FCR 549 (Full Federal Court); *Computer Edge Pty Ltd v Apple Computer Inc* (1986) 161 CLR 171 (High Court).

⁶⁰ *Computer Edge v Apple* citing in support *Hollinrake v Truswell* [1894] 3 Ch. 420; *Exxon Corporation v Exxon Insurance Consultants International Ltd* [1982] R.P.C. 69 at 88.

Is programming a fine art or a useful art?

On appeal to the Full Federal Court, source code was held to “express meaning as to the arrangement and ordering of instructions for the storage and reproduction of knowledge. It is incorrect to describe them simply as components of a machine.”⁶¹ The Full Court held that:

[t]here is no necessity for a literary work to be of any literary quality. It is accepted that the term includes mathematical tables, codes, and, in general, alphanumerical works. One limit doubtless is that it needs to be a “work” and to have had some skill, even if very small, applied to its preparation. Meaningless rubbish would plainly be excluded.⁶²

The Full Court took the view that it did not matter whether object code was a literary work or not, “because [it] can fairly be described as [a] translation[.]”⁶³ On further appeal, the High Court reinforced the position, holding that source code was akin to a literary work, but object code was not:

[The object] programs existed in the form of a sequence of electrical impulses, or possibly in the pattern of circuits that when activated generated those electrical impulses. On any view they were not expressed in writing or print. Although the electrical impulses could be represented by words or figures, the impulses themselves did not represent or reproduce any words and figures. They were not visible or otherwise perceptible, and they were not, and were not intended to be, capable by themselves of conveying a meaning which could be understood by human beings. ... It is true that the object programs might have been printed out in binary or hexadecimal form, but the question whether any such written expression of the programs would have been a literary work is not the question that now falls for decision. We are concerned with the object programs embodied in the ROMs and it seems clearly to follow from the cases already cited, which decide that a literary work is a work expressed in print or writing, that they were not literary works.⁶⁴

Thus it might be argued that the tendency is to emphasise the expressiveness as located in source code, and not present in object code, which is functional. The solution in Australia has been to amend the *Copyright Act 1968* to expressly include object code within the definition of a literary work.

The US position, which proceeds from a similar theoretical position,⁶⁵ provides an interesting counterpoint. For example, in *Williams Electronics v Artic International*⁶⁶ the defendant to a claim for infringement asserted just such a distinction between expressive source code and functional object code. However, the court declined to draw such a distinction, holding that object code

⁶¹ *Apple Computer v Computer Edge* (1983) 1 FCR 549 at 558-559 per Fox J.

⁶² *Apple Computer v Computer Edge* (1983) 1 FCR 549 at 558 per Fox J.

⁶³ *Apple Computer v Computer Edge* (1983) 1 FCR 549 at 559 per Fox J.

⁶⁴ *Computer Edge Pty Ltd v Apple Computer Inc* (1986) 161 CLR 171 at 183 per Gibbs CJ.

⁶⁵ So to the UK position. See *SAS Institute Inc v World Programming Ltd* [2010] EWHC 1829 (Ch) in which it was held that the WIPO treaty, Berne Convention and TRIPS all support the distinction between copyrightable expression and unprotectable ideas.

⁶⁶ *Williams Electronics v Artic International* 685 F.2d 870 (3rd Circuit, 1986) at 876-877

Chapter 7: Why Programming is not among the Useful Arts

came within the terms of the US Copyright Act.⁶⁷ This is in no small part due to the legislative definition by which a “copy”

include[s] a material object in which a work is fixed “by any method now known or later developed, and from which the work can be perceived, reproduced, or otherwise communicated, either directly or with the aid of a machine or device.” By this broad language, Congress opted for an expansive interpretation of the terms “fixation” and “copy” which encompass technological advances such as those represented by the electronic devices in this case.⁶⁸

The Court in *Apple v Franklin*⁶⁹ approached the same issue from a slightly different angle, invoking the *Baker v Selden* case as authority for distinguishing between copyright protection for the “mere” instructions as a literary work on the one hand, and protecting the method behind the instructions as a utilitarian, or functional, work. The issue in patent law is perhaps reversed, since it is asserted that expressiveness is an indicium of a non-patentable fine art. To that end, it is worth exploring further the extent to which all code, even the executable code most aligned with function, achieves a level of expressiveness.

A starting point in assessing the expressiveness of code is to say something more about the medium in which software is constructed – the programming *language*. A non-programmer might be tempted by the view that such “languages” are no more than convenient standards by which to interface to, or direct, the computer’s machine-code operations.⁷⁰ Put another way, a language might be seen as nothing more than a “fixed collection of rules”⁷¹ to be assembled in a mechanical manner to achieve the required function. If these languages were mere “standards” we might expect that there were only a few of them, in much the same way that operating systems have “standardised” around two major variations – Windows and Unix.⁷²

But a computer language is much more than an objective standard. It is “a system of notation for describing computation”⁷³ with an underlying structure that reflects the language designer’s decisions.⁷⁴ These decisions are a compromise between conflicting goals:

A useful programming language must ... be suited for both description (i.e., for human writers and readers of programs) and for computation (i.e., for efficient implementation on computers). But human beings and computers are so different

⁶⁷ Copyright Act 17 USC (1976).

⁶⁸ *Williams Electronics v Artic International* 685 F.2d 870 (3rd Circuit, 1986) at 876-877.

⁶⁹ 714 F.2d 1240 (3rd circuit, 1983).

⁷⁰ “Business types prefer the most popular languages because they view languages as standards.”: Paul Graham “Great Hackers” <<http://www.paulgraham.com/gh.html>> (16 June 2011).

⁷¹ Hal Abelson, “Foreword” in Daniel P Friedman, Mitchell Wand and Christopher T Haynes, *Essentials of Programming Languages* (MIT Press, 2001) at vii.

⁷² This reductionist view might itself be challenged. Firstly, Unix includes a large number of variants, including a number of GNU/Linux and BSD distributions, Solaris and MacOSX. Secondly, although Unix variants share a design, Windows variants only share a trade mark, and a certain degree of backward compatibility with earlier brand members.

⁷³ Robert D Tennent, *Principles of Programming Languages* (Prentice Hall, 1990) at 1. See also *Junger v Daley*

⁷⁴ Hal Abelson, “Foreword” in Daniel P Friedman, Mitchell Wand and Christopher T Haynes, *Essentials of Programming Languages* (MIT Press, 2001) at vii.

that it is difficult to find notational devices that are well suited to the capabilities of both.⁷⁵

The scope for variation between the formal specification of a particular language, (discussed below in relation to Python and C#) also suggests the appropriateness of analogies with other expressive works, such as literary and other copyrightable works. Just as there are multiple ways to write a book given a specific storyline,⁷⁶ there are myriad ways to give effect to the function of an algorithm, and programmers at least see value in exploring different expressions. If a language is merely a standard, surely there would be no need for more than one implementation of it. This is suggestive of the value which lies in the expression of computable processes in a variety of ways.⁷⁷

Also influencing language design is the fact that programmers have their own styles that reflect the way they think,⁷⁸ and the capacity of languages to influence the way programmers think makes language choice an important decision.⁷⁹ For example, the Perl language is designed around the idea that “there’s more than one way to do it”.⁸⁰ In contrast, the Python language’s philosophy is that “there should be one– and preferably only one – obvious way to do things”.⁸¹

What these influences on language design also demonstrate, is that a programming language may be thought of as just another program. In fact, Abelson describes this very notion as “the most fundamental idea in computer programming”:⁸²

⁷⁵ Robert D Tennent, *Principles of Programming Languages* (Prentice Hall, 1990) at 1.

⁷⁶ In fact, it is suggested at various times that there are only a very small number of story lines to choose from, from 1 to 30. See IPL2, “The ‘Basic’ Plots in Literature” <<http://www.ipl.org/div/farq/plotFARQ.html>> (8 July 2011).

⁷⁷ The notion of value in programming is explored below in Section 2 on page 292.

⁷⁸ “[Language] syntax deeply affects how you think about the problem, even though semantically it has absolutely no bearing on what’s going on.” Anders Hejlsberg, designer of the C# language, in John Osborn, “C#: Yesterday, Today, and Tomorrow: An Interview with Anders Hejlsberg, Part 2” *OnDotNet*, 31 October 2005 <<http://ondotnet.com/pub/a/dotnet/2005/10/31/interview-with-anders-hejlsberg-part-2.html>> (8 July 2011).

⁷⁹ Dijkstra was an ardent proponent of the influence of computer languages on thought:

The tools we use have a profound (and devious!) influence on our thinking habits, and, therefore, on our thinking abilities.

...

It is practically impossible to teach good programming to students that have had a prior exposure to BASIC: as potential programmers they are mentally mutilated beyond hope of regeneration.

The use of COBOL cripples the mind; its teaching should, therefore, be regarded as a criminal offence.”

Edsger W Dijkstra, “How do we tell truths that might hurt?”, (EWD498, 1975)

<<http://www.cs.utexas.edu/~EWD/transcriptions/EWD04xx/EWD498.html>> (29 July 2011), reproduced in *Selected Writings on Computing: A Personal Perspective* (Springer-Verlag) at 129-131. See also Robin Milner, designer of the ML language, cited in Dexter et al at 12: “[S]ome languages ... actually influence the way that the programmer thinks about the task. Object-oriented languages have done very well from this viewpoint, because the notion of object helps to clarify thought in a remarkable variety of applications”. The relationship between expression and thought was discussed in Chapter 5.

⁸⁰ Wikipedia, “There’s more than one way to do it”

<http://en.wikipedia.org/wiki/There's_more_than_one_way_to_do_it> (11 July 2011).

⁸¹ Tim Peters, “PEP 20 – The Zen of Python” <<http://www.python.org/dev/peps/pep-0020/>> (7 July 2011). This list of attributes of the python language can be accessed from the python interpreter itself by issuing the command “import this”. The inventor of the language, and lead programmer, Guido van Rossum, when asked about the origin of this design goal, noted that it “comes straight from the general desire for elegance in mathematics and computer science”, being an example of orthogonality: Frederico Biancuzzi and Shane Warden, *Masterminds of Programming* (O’Reilly Media, 2009) at 25.

⁸² Hal Abelson, “Foreword” in Daniel P Friedman, Mitchell Wand and Christopher T Haynes, *Essentials of Programming Languages* (MIT Press, 2001) at vii.

Chapter 7: Why Programming is not among the Useful Arts

Consider ... the basic idea: the interpreter itself is just a program. But that program is written in some language, whose interpreter is itself just a program written in some language whose interpreter is itself...⁸³

This recursiveness can be demonstrated by looking at the Python language. On a purely abstract level, Python can be thought of as a mere collection rules of syntax and semantics.⁸⁴ Giving effect to that abstract specification is a number of implementations. The “main” implementation of Python is written in C, and is sometimes referred to as CPython. There are also implementations written in Java (Jython)⁸⁵ and C# (IronPython).⁸⁶ There is even an implementation written in Python itself (PyPy).⁸⁷ Digging a level deeper, the IronPython interpreter is written in C#, a language developed by Microsoft, which is written in C++,⁸⁸ and compiles to an intermediate language called MSIL.⁸⁹ MSIL can be compiled to machine code beforehand, or executed by a machine-code interpreter at run-time, these compilers/interpreters are “likely to have been written in Microsoft Visual C”.⁹⁰ To actually run, the executable object code depends on a framework called the Common Language Framework, which was written in C++, although it seems that at least one component was prototyped in Lisp.⁹¹ The Microsoft specifications for C#, MSIL and machine code interpreters have been independently implemented by an open source project called Mono, which is entirely implemented in C#.⁹² Mono is available for Mac OSX, Windows and various Linux distributions.⁹³

Putting aside the distraction of all the technical details, this short overview serves to demonstrate both how a language is a program, assembled from a series of component parts, in much the same way as any modern application can be constructed from the building blocks of a language, various libraries and frameworks.⁹⁴

⁸³ Hal Abelson, “Foreword” in Daniel P Friedman, Mitchell Wand and Christopher T Haynes, *Essentials of Programming Languages* (MIT Press, 2001) at viii.

⁸⁴ Although there is no rigorous formal specification of every aspect of the language, the Python Language Reference is considered “the” guide. See “The Python Language Reference” <<http://docs.python.org/reference/index.html>> (11 June 2011).

⁸⁵ See <<http://www.jython.org/>> (11 June 2011).

⁸⁶ <<http://ironpython.net>> (8 July 2011).

⁸⁷ <<http://pypy.org/>> (11 June 2011). Implementing a language in itself is known as bootstrapping, or self-hosting. “A large proportion of programming languages are bootstrapped, including BASIC, C, Pascal, Factor, Haskell, Modula-2, Oberon, OCaml, Common Lisp, Scheme, Python and more.”: Wikipedia, “Bootstrapping (compilers)” <[http://en.wikipedia.org/wiki/Bootstrapping_\(compilers\)](http://en.wikipedia.org/wiki/Bootstrapping_(compilers))> (27 May 2011).

⁸⁸ At the very least, the shared source implementation of C# is. See “Shared Source Common Language Infrastructure 2.0 Release” <<http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=4917>> (8 July 2011).

⁸⁹ This stands for Microsoft Intermediate Language. See Microsoft, “Managed Execution Framework” <<http://msdn.microsoft.com/en-us/library/k5532s8a.aspx>> (8 July 2011).

⁹⁰ Because this is a proprietary program, the source code is not available. However, the likelihood of the compiler, ngen.exe having been written in C is asserted in FaultWire’s analysis. See FaultWire, “Ngen.exe CLR Native Compiler” <https://www.faultwire.com/file_detail/ngen.exe*41826.html> (8 July 2011).

⁹¹ “I designed the architecture of the runtime and wrote the Garbage Collector (and yes the GC prototype was written in Common Lisp first and I wrote a translator to convert it to C++)”: Patrick Dussud, “How It All Started... AKA the Birth of the CLR” on CLR, *Architectures and Stuff* <http://blogs.msdn.com/b/patrick_dussud/archive/2006/11/21/how-it-all-started-aka-the-birth-of-the-clr.aspx> (8 July 2011).

⁹² “The [Mono C#] compiler is routinely used to compile Mono, roughly four million lines of C# code and a few other projects.”: Mono Project, “CSharp Compiler” <http://www.mono-project.com/CSharp_Compiler> (8 July 2011).

⁹³ See <<http://www.go-mono.com/mono-downloads/download.html>> (8 July 2011).

⁹⁴ See the discussion in Chapter 1 at 17.

Is programming a fine art or a useful art?

The example also demonstrates how the division between source code and object code is not cleanly drawn. Ultimately, the form into which the source code is translated *is* one in which it is capable of actually causing the computer to carry out the described process. But that path to a final form “a complex and remarkable chain of events”,⁹⁵ and may not be completed until just before the time the code is executed. Any claims to a clean separation between expressive source code and functional object code must therefore be cautiously approached.

It also demonstrates how code can be seen to be *executable data* – the source code for software, rather than being seen as a mere adjunct to the process as executing, can be understood as the precise description of a computable process. That description forms the input (data) to a program which translates it to another form. The output of that program may be into machine code, or it may be to another intermediate language requiring further translation before execution. The software also may not reach its final, executable state until just before it is in fact executed.

An important corollary of the interpreter as a program should also be considered. Any program can also be conceived of as a language, a collection of languages, or at the very least as containing “important [language]-like pieces”.⁹⁶ Larry Wall, designer of the Perl programming language, adverts to this when he suggests that “if you think of Unix as a programming language, it’s far richer than even Perl. Perl is, by and large, a digested and simplified version of Unix. Perl is the Cliff Notes of Unix.”⁹⁷ Abelson advocates exactly such an understanding, suggesting that

[o]ne of the most powerful ways to structure a complex program is as a collection of languages, each of which provides a different perspective, a different way of working with the program elements.⁹⁸

In the absence of a clear division between source code and object code, Colburn and Shute suggest a way of characterising the expressiveness of software as a function of the level of abstraction involved:

As levels of programming language abstraction increase, the languages become more expressive in the sense that programmers can manipulate direct analogs of objects that populate the world they are modeling, like shopping carts, chat rooms, and basketball teams. This expressiveness is only possible by hiding the complexity of the interaction patterns occurring at lower levels.⁹⁹

⁹⁵ Timothy Colburn and Gary Shute, “Abstraction, Law and Freedom in Computer Science” (2010) 41(3) *Metaphilosophy* 345 at 355.

⁹⁶ Hal Abelson, “Foreword” in Daniel P Friedman, Mitchell Wand and Christopher T Haynes, *Essentials of Programming Languages* (2nd ed, MIT Press, 2001) at viii.

⁹⁷ Larry Wall, “Perl, the first postmodern computer language” <<http://www.perl.com/pub/1999/03/pm.html>> (7 July 2011).

⁹⁸ Hal Abelson, “Foreword” in Daniel P Friedman, Mitchell Wand and Christopher T Haynes, *Essentials of Programming Languages* (2nd ed, MIT Press, 2001) at viii.

⁹⁹ Timothy Colburn and Gary Shute, “Abstraction, Law and Freedom in Computer Science” (2010) 41(3) *Metaphilosophy* 345 at 348.

Chapter 7: Why Programming is not among the Useful Arts

This suggests that expressiveness is determined by physicality, being proportional to the extent to which physical restrictions impose themselves on the programming process. Such a view certainly lends further weight to the analysis set out in relation to physicality above, and in particular the need for careful characterisation of the invention.

In this context it may also be useful to determine the baseline: how expressive is the most functional of all code, that is, code in its final machine-executable form? In Chapter 4 it was asserted that the “new machine” characterisation of software, whilst correct as a matter of computer science, is nevertheless problematic.¹⁰⁰ One reason for this is that such a characterisation at best overlooks, and at worst strips away, an essential characteristic of software:

Because the operations of these machines have no apparent meaning outside of human activity, it is easy to conclude ... that the meaning of a program is identical to the human-interpreted result of its function: once the ‘internal’ mental symbols are brought outside the body, the crucial connection with ‘external’ states is severed, the ‘internal’ mental symbols all but vanish as they are stripped of their meaning, and we are left with machines which, with a compelling illusion of near-autonomy, traverse a wide range of meaning-laden states.¹⁰¹

Put another way, software is not adequately described by its mere functionality, “existing outside of and independent from human experience”¹⁰² but is inextricably linked to its human meaning. There is a temptation to dismiss such an argument on the basis that any technical or engineering process is imbued with just such a kernel of expression. Expressivity can operate to positively enhance the perception of functionality. A Mac fanboi¹⁰³ purchases a PowerBook not only because of its technical specifications, like processor speed and memory capacity, but because of what it *means* to be a Mac owner. It might be a statement about valuing Apple’s “it just works” philosophy,¹⁰⁴ a commitment to industrial design, or even a protest against the dominance of Windows in the operating systems market.

This grounding of meaning in what might at first blush be thought to be predominantly functional opens the door to an understanding of the ways in which software is “technically expressive”. Ratto demonstrates how software may contain of “embedded technological ex-

¹⁰⁰ See Chapter 4, at 164.

¹⁰¹ Scott Dexter et al, “On the Embodied Aesthetics of Code” (2011) 12 *Culture Machine* 1 at 10.

¹⁰² Scott Dexter et al, “On the Embodied Aesthetics of Code” (2011) 12 *Culture Machine* 1 at 9.

¹⁰³ A fanboi is a pejorative term for “a person with an irrational attachment to a particular item or brand name, and an equally irrational dislike for competing brands or items.”: “fanboi or fanbois”, *Whirlpool Knowledge Base* <<http://whirlpool.net.au/wiki/fanboi>> (4 August 2011). See also “fanboi” *Urban Dictionary* <<http://www.urbandictionary.com/define.php?term=fanboi>> (4 August 2011). For a report of a study linking fanbois behaviour to threats to self image, see Jacqui Cheng, “Fanbois treat criticism of favorite brands as threat to self-image” *Ars Technica*, 17 August 2011 <<http://arstechnica.com/science/news/2011/08/users-treat-criticism-of-favorite-brands-as-threat-to-self-image.ars>> (18 August 2011).

¹⁰⁴ That Macs just work is an oft-repeated claim: “Enter the Mac – I plugged the drive into my G4 and it worked immediately. This is what makes Mac OS X so compelling: With [sic] decimated IT staffs getting slammed from without (Sobig.F) and within (frustrated end-users), something that just works is downright refreshing.”: Chad Dickerson, “Mac OSX: it just works” *Infoworld*, 12 September 2003 <<http://www.infoworld.com/t/platforms/mac-os-x-it-just-works-441>> (4 August 2011). The phrase was originally an Apple slogan for its Unix-based operating system, OSX. See Wikipedia, “List of Apple Inc. slogans” <http://en.wikipedia.org/wiki/List_of_Apple_Inc._slogans> (4 August 2011).

pression"¹⁰⁵ in three ways. Firstly, as touched on above, the source code of a program expresses, or explains, the process whereby particular tasks can be accomplished.¹⁰⁶ Secondly, software when run on a computer expresses to users how they may interact with the software.¹⁰⁷ Thirdly, the modularisation of a program undertaken as a part of the design process acts as a method of organising labour. That is, it defines "relationships between programmers".¹⁰⁸

Dijkstra believed that expression is central to the activity of programming, although in a way which extends both beyond the mere expression of a particular idea in a programming language, and the notions of technical expression set out above. Dijkstra believed the need for natural language mastery in programming was twofold: firstly, it was necessary for the accurate communication and description of the problem to be solved, and also as an "indispensible tool for thinking".¹⁰⁹

Summary

The classical approach to determining the appropriate protection for software is argued along this dimension. On the one hand, copyright is said to protect expressive works, and patent law to protect functional works. However programs are "machines whose medium of construction is text"¹¹⁰ and exhibit a duality or deep interdependence of functionality and expressiveness which makes it hard to classify.

Attempts to shoehorn software into one or other category have therefore faced difficulties. In the copyright context, the expressive nature of source code is emphasised, as being akin to a literary work. However, in recognition of the fact that there is a tight causal relationship between source code and object code, copyright regimes have been expanded to cover object code, which shares little similarity with other copyrightable works, in that it *behaves*.

Similarly, in the patent law context, a focus on the function of software has a similarly damaging effect. There is no doubt that software is functional. To assert otherwise would be folly.

¹⁰⁵ Dan L Burk, "Patenting speech," (2000) 79 *Texas Law Review* 99 at 115; Matt Ratto, "Leveraging Software, Advocating Ideology: Free Software and Open Source" paper presented to the 29th Research Conference on Information, Communication and Internet Policy (2001) <<http://arxiv.org/abs/cs.CY/0109103>> (27 June 2011) at 12.

¹⁰⁶ Matt Ratto, "Leveraging Software, Advocating Ideology: Free Software and Open Source" paper presented to the 29th Research Conference on Information, Communication and Internet Policy (2001) <<http://arxiv.org/abs/cs.CY/0109103>> (27 June 2011) at 12.

¹⁰⁷ "[S]oftware programs express appropriate methods of doing tasks to users, while not necessarily constraining user behavior": Matt Ratto, "Leveraging Software, Advocating Ideology: Free Software and Open Source" paper presented to the 29th Research Conference on Information, Communication and Internet Policy (2001) <<http://arxiv.org/abs/cs.CY/0109103>> (27 June 2011) at 14.

¹⁰⁸ "[S]tructure refers to the organization of different code elements within the overall software program... Such organizing code structures also organize the labor that goes into coding, maintaining, or extending programs. In this way, code structure expresses relationships between programmers.": Matt Ratto, *Leveraging Software, Advocating Ideology: Free Software and Open Source* at 157. The relationship between organisational structure and the structure of code is also reflected in Conway's Law, that "[a]ny organization that designs a system will produce a design whose structure is a copy of the organization's communication structure.... you can see the idea in the writings of Eric Evans and Jim McCarthy": <http://beautifulcode.oreillynet.com/2007/11/hidden_influences_on_software.php> (15 July 2008).

¹⁰⁹ E. W. Dijkstra "Programming as a Discipline of Mathematical Nature" (1974) 81(6) *The American Mathematical Monthly* 608-612.

¹¹⁰ Pamela Samuelson et al, "A Manifesto Concerning the Legal Protection of Computer Programs," (1994) 94 *Columbia Law Review* 2308 at 2320.

Chapter 7: Why Programming is not among the Useful Arts

However, software is expressive, in at least the three ways just described. So if anything, the choice in this context might be between software as functionally expressive or expressively functional. Given the way in which it has been demonstrated that software's function is also expressive, the latter of these terms would seem the most appropriate.

The result of allowing patenting of subject matter which arguably strays into the realm of the expressive has meant the traditional limitations on patent law have been watered down to the point where the regime has allowed patents over movie scripts, sports moves and tax advice. This fact may be suggestive of the appropriateness of finding a solution to the software patent problem which is reflective of its unique character – a *sui generis* regime.

Aesthetic or Rational?

Looking at the design process for software, and recalling the differences attributed to artistic and engineering design by Mitcham by reference to their “ends in view”,¹¹¹ it is possible to determine whether software is primarily directed to efficiency or beauty. That is, whether it is a rational or aesthetic creation. This may be a question of degree to some extent, since it is clear that even in traditional engineering disciplines such as building of bridges and towers, “structural engineers are guided by aesthetics as well as [rational] analysis”.¹¹² However, whilst both aesthetic and rational considerations might be at play in software development, it is submitted that the aesthetic aspect is so integral to software development that it cannot be considered a mere ancillary factor, as it might be in other technological endeavours.

A rational engineering approach involves a sequence of logical steps. A good example of this model in action is the idealized waterfall model of software development, set out in Figure 7.1. This approach to software development dominated in the early years of software development, and was a product of the dominance of hardware engineering, and hence a hardware engineering approach.¹¹³ Under the waterfall model, each phase of the design and development process ought to be completed before the next is begun, with some small tolerance for interaction between levels.

But this ideal of a rational design process has never matched the reality of software development, and never will. At best, it will be seen that this model is something which might be worth “faking”,¹¹⁴ in the same way that the rigorous logical deduction from axiom to proof in a published mathematical proof belies the tortuous and winding path of discovery which precedes it.¹¹⁵ This “tortuous path” is the hallmark of the irrational, emotional, intuitive, hu-

¹¹¹ See Chapter 5, at 5.

¹¹² Bruce J MacLennan, “Aesthetics in Software Engineering” (Technical Report No UT-CS-06-579, Department of Electrical Engineering and Computer Science, University of Tennessee, 31 October 2006) <<http://web.eecs.utk.edu/~library/TechReports/2006/ut-cs-06-579.pdf>> (26 May 2011).

¹¹³ Barry Boehm, “A View of 20th and 21st Century Software Engineering” (Paper presented at the 28th International Conference on Software Engineering, Shanghai, China, 20 May 2006) 12 <<http://portal.acm.org/citation.cfm?id=1134288>> at 13.

¹¹⁴ David L Parnas and Paul C Clements, “A Rational Design Process: How and Why To Fake It” (1986) 12(2) *IEEE Transactions on Software Engineering* 251.

¹¹⁵ David L Parnas and Paul C Clements, “A Rational Design Process: How and Why To Fake It” (1986) 12(2) *IEEE Transactions on Software Engineering* 251 at 251: “[T]o many observers, the usual process of designing software

Is programming a fine art or a useful art?

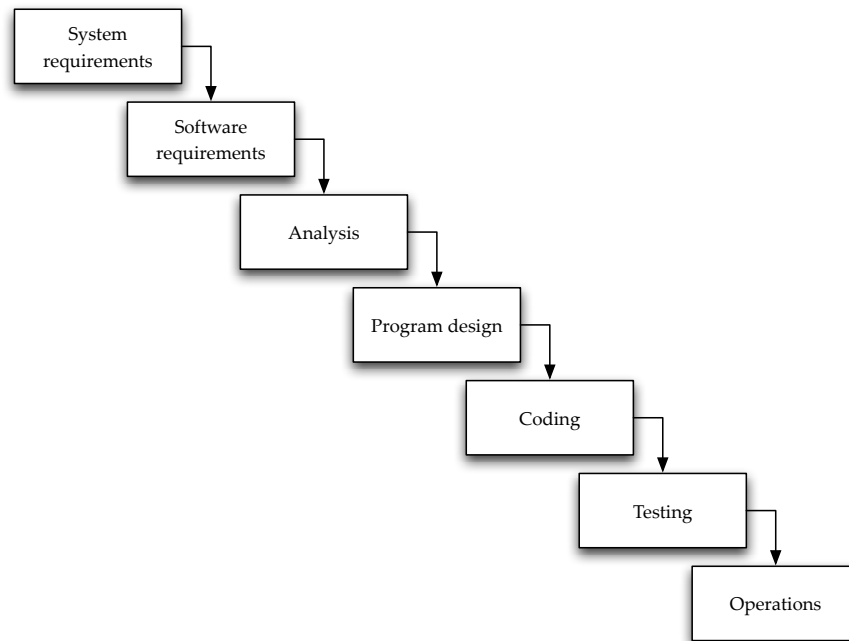


Figure 7.1: The idealised waterfall development model

man, or aesthetic mode of creation. It is how books are written, how a block of stone becomes a sculpture, how an artist moves from an idea to a work in a gallery.

Just another type of engineering?

The development of software might at first blush seem primarily rational. In many cases “the [software development] problem is appropriately an engineering one: creating cost-effective solutions to practical problems, building things in the service of mankind.”¹¹⁶ It is directed towards the assembly of a sequence of logical steps, syntactically and grammatically correct according to the strictures of the “language” in which it is written. Not only must it in fact be correct in order to compile and run, thereby transforming the computer into a “new machine”,¹¹⁷ but the eventual “product” will be expected to efficiently utilise the available re-

appears quite irrational. Programmers start without a clear statement of desired behaviour and implementation constraints. They make a long sequence of design decisions with no clear statement of why they do things the way they do. Their rationale is rarely explained.” See also Daniel M Berry, “The Inevitable Pain of Software Development: Why There Is No Silver Bullet” in Martin Wirsing, Alexander Knapp and Simonetta Balsamo (eds), *Radical Innovations of Software and Systems Engineering in the Future, Proceedings of the 2002 Monterey Conference*, LNCS 2941 (Springer, 2004) 50 at 53: “No matter how much we try to be systematic and to document what we are doing, we forget to write things down, we overlook some things, and the discoveries seem to grow faster than the code.” As Charles Strauss notes, this unusual process can be observed at work in programmers. “Mostly, when you see programmers, they aren’t doing anything. One of the attractive things about programmers is that you cannot tell whether or not they are working simply by looking at them. Very often they’re sitting there seemingly drinking coffee and gossiping, or just staring into space. What the programmer is trying to do is get a handle on all the individual and unrelated ideas that are scampering around in his head.”: Philip J Davis and Reuben Hersh *Descartes’ Dream; The World According to Mathematics* (Dover Publications, 1986) at 180.

¹¹⁶ Mary Shaw, “Prospects for an Engineering Discipline of Software” (1990) 7(6) *IEEE Software* 15, at 19.

¹¹⁷ For a discussion of the new machine characterisation of software, see Chapter 4, at 164.

Chapter 7: Why Programming is not among the Useful Arts

sources of the computer to perform the functions which it was written to perform. As noted by Mitcham, this notion of efficiency is at the heart of engineering design.¹¹⁸

Also, many programmers are likely to see themselves as engineers, working through a series of rational steps towards a defined end goal. This link with engineering has a historical basis. When software first emerged in the 1950s it was, understandably, closely aligned with computer hardware engineering.¹¹⁹ In the 1960s however, the demand for software coupled with the emerging “hacker” culture saw software development break free of this close association with hardware engineering, into a discipline in its own right.¹²⁰ At the same time, the growing complexity of software projects, the increasing ambition of programmers, and a series of colossal failures on software development projects prompted NASA to hold two conferences on the emerging “software crisis”.¹²¹ In response to this crisis, the term “software engineering” was coined in 1968.¹²² Software engineering, like any other engineering discipline, would need to proceed on a disciplined, rational basis, in order to achieve the productivity gains comparable to those being achieved by computer hardware engineers.¹²³

This view of software development as a rational process might also be reinforced by the fact that software development is an industry of enormous economic importance. Software products keep getting bigger, more complex, and require better organisation and management in order to be developed. Organisations developing software solutions therefore stand to benefit from reinforcing the stable, “engineering” nature of software development in order to assure customers and investors that they are making a safe bet.¹²⁴ However, it is submitted that such project management and business process issues lie beyond any properly defined software en-

¹¹⁸ Cf H Coqui, “Corporate survival: The software dimension” Focus ’89, Cannes, 1989, cited in Frederick P Brooks Jr, *The Mythical Man Month: Essays on Software Engineering* (Anniversary ed, Addison-Wesley, 2002) at 218: “The driving force to use Software Engineering principles in software production was the fear of major accidents that might be caused by having uncontrollable artists responsible for the development of ever more complex systems.”

¹¹⁹ “When I entered the software field in 1955 ... the prevailing thesis was, ‘Engineer software like you engineer hardware’. Everyone in the ... organization was a hardware engineer or a mathematician, and the software being developed was supporting aircraft or rocket engineering. People kept engineering notebooks and practiced such hardware precepts as ‘measure twice, cut once,’ before running their code on the computer.”: Barry Boehm, “A View of 20th and 21st Century Software Engineering” (Paper presented at the 28th International Conference on Software Engineering, Shanghai, China, 20 May 2006) 12 <<http://portal.acm.org/citation.cfm?id=1134288>> at 13.

¹²⁰ Barry Boehm, “A View of 20th and 21st Century Software Engineering” (Paper presented at the 28th International Conference on Software Engineering, Shanghai, China, 20 May 2006) 12 <<http://portal.acm.org/citation.cfm?id=1134288>> at 13-14.

¹²¹ Barry Boehm, “A View of 20th and 21st Century Software Engineering” (Paper presented at the 28th International Conference on Software Engineering, Shanghai, China, 20 May 2006) 12 <<http://portal.acm.org/citation.cfm?id=1134288>> at 14. Dijkstra asserted that the crisis was directly referable to the nature of the programming activity: “The major cause of the software crisis is that the machines have become several orders of magnitude more powerful! To put it quite bluntly: as long as there were no machines, programming was no problem at all; when we had a few weak computers, programming became a mild problem, and now we have gigantic computers, programming has become an equally gigantic problem.”: Edsger W Dijkstra, “The Humble Programmer”, (1972) 15 (10) *Communications of the ACM* 859 at 862.

¹²² Barry Boehm, “A View of 20th and 21st Century Software Engineering” (Paper presented at the 28th International Conference on Software Engineering, Shanghai, China, 20 May 2006) 12 <<http://portal.acm.org/citation.cfm?id=1134288>> at 14.

¹²³ Frederick P Brooks Jr, “No Silver Bullet: Essence and Accidents of Software Engineering” (1987) 20(4) *Computer* 10 at 10.

¹²⁴ Such an approach is no doubt behind Ershov’s lamentation in 1972 that “authority over the freewheeling brotherhood of programmers is slipping into the paws of administrators and managers – who try to make the work of programmers planned, measurable, uniform and faceless.”: Andrei P Ershov, “Aesthetics and the Human Factor in Programming” (1972) 15(7) *Communications of the ACM* 501.

gineering scope.¹²⁵

No Silver Bullet

It was noted that the historical schism between programming and hardware engineering emerged in the 1960s. Despite the attempts to recharacterise software development as an engineering discipline, in 2009, Shaw acknowledged that no practice of software engineering had yet emerged.¹²⁶ Whilst new methodologies are regularly said to herald the dawn of the software engineering age,¹²⁷ these so-called “silver bullets”¹²⁸ always miss their mark.

Further, there are good reasons to think that a true engineering practice in software development may never take hold. The main reason for adopting such a negative view stems from the inherent nature of the software development task. Engineering practice limits the impact of variance between individuals, by enabling “ordinary practitioners so they can create sophisticated systems that work – unspectacularly perhaps, but reliably.”¹²⁹ Where an activity is objective, or rational, this is easily done. But “[t]he essence of a software entity is a construct of interlocking concepts; data sets, relationships among data items, algorithms and invocations of functions.”¹³⁰ As such, “the hard part of building software [is] the specification, design and testing of this conceptual construct, not the labor of representing it and testing the fidelity of the representation.”¹³¹ This distinguishes software practice from engineering practice:

Whereas the difference between poor conceptual designs and good ones may lie in the soundness of design method, the difference between good designs and great ones surely does not.¹³²

¹²⁵ “[W]henver someone says ‘we don’t need technical advances; we need a better process,’ that’s a sign that production skills haven’t yet brought us to a fully mature commercial practice.”: Mary Shaw, “Continuing Prospects for an Engineering Discipline of Software” (2009) 26(6) *IEEE Software* 64 at 66. According to Shaw, a mature commercial practice is a pre-cursor to the emergence of an engineering practice.

¹²⁶ Mary Shaw, “Continuing Prospects for an Engineering Discipline of Software” (2009) 26(6) *IEEE Software* 64.

¹²⁷ In the patent law context, Drjya for example argues that the move away from the Turing programming model, which “caused” the software crisis, towards the object-oriented programming paradigm has seen a distinct change in the nature of software development from art to engineering. See Michael A Dryja, “Looking to the Changing Nature of Software Development for Clues to Its Protection” (1995) 3 *University of Baltimore Intellectual Property Law Journal* 109. Shaw takes a positive, yet more guarded position, arguing that there “are good grounds to expect that there will eventually be an engineering discipline of software.”: Mary Shaw, “Prospects for an Engineering Discipline of Software” (1990) 7(6) *IEEE Software* 15 at 22.

¹²⁸ “Of all the monsters that fill the nightmares of our folklore, none terrify more than werewolves, because they transform unexpectedly from the familiar into horrors. For these, one seeks bullets of silver that can magically lay them to rest.

The familiar software project, at least as seen by the nontechnical manager, has something of this character; it is usually innocent and straightforward, but is capable of becoming a monster of missed schedules, blown budgets, and flawed products. So we hear desperate cries for a silver bullet – something to make software costs drop as rapidly as computer hardware costs do.”: Frederick P Brooks Jr, “No Silver Bullet: Essence and Accidents of Software Engineering” (1987) 20(4) *Computer* 10 at 10.

¹²⁹ Mary Shaw, “Prospects for an Engineering Discipline of Software” (1990) 7(6) *IEEE Software* 15 at 16.

¹³⁰ Frederick P Brooks Jr, “No Silver Bullet: Essence and Accidents of Software Engineering” (1987) 20(4) *Computer* 10 at 11.

¹³¹ Frederick P Brooks Jr, “No Silver Bullet: Essence and Accidents of Software Engineering” (1987) 20(4) *Computer* 10 at 11.

¹³² Frederick P Brooks Jr, “No Silver Bullet: Essence and Accidents of Software Engineering” (1987) 20(4) *Computer* 10 at 18.

Chapter 7: Why Programming is not among the Useful Arts

Correspondingly, there is some empirical evidence to suggest that programmers of similar experience may exhibit differences in productivity of an order of magnitude.¹³³ As an endeavour dependent on individual attributes, it is submitted that aesthetic considerations must play a central role.

Beynon et al note that “[o]ne of the main purposes of adopting a formal approach to program development is to shift the focus from ‘considering your intuitive ideas about what a program is intended to do’ to ‘thinking precisely and abstractly about what is it that you have instructed it to do’.”¹³⁴ Viewed as such, emphasising such a structured approach to software development address the potential hazard which exists “in the relation between what we can see immediately, and perhaps superficially, and what we might be able to see upon closer scrutiny in other contexts, possibly unforeseen.”¹³⁵

However, a formal approach, whilst perhaps an important teaching tool, does not remove intuition, which may still operate in the interpretation of a formal specification.¹³⁶ Further, a formal approach does not guarantee success in software development, and is only one possible way of approaching the software development task.¹³⁷

Further, whilst large software development houses such as Microsoft, Adobe and Oracle might have a prominence in the marketplace, not all software is developed by such large software houses. Often, smaller development teams are involved. Further, as Brooks notes, the “genius” often arises from the work of individual developers.¹³⁸ Similarly, it is not the large-scale software architectures which form the basis of patent applications, but instead the granular software components. So a focus on the software product, and the organisation which produces it, overlooks that this is basically a human activity.

¹³³ H Sackman, W J Erickson and E E Grant, “Exploratory Experimental Studies Comparing Online and Offline Programming Performance” (1968) 11(1) *Communications of the ACM* 3; Bill Curtis, “Fifteen Years of Psychology in Software Engineering: Individual Differences and Cognitive Science” ICSE ’84 Proceedings of the 7th international conference on Software engineering (IEEE Press, 1984) <<http://portal.acm.org/citation.cfm?id=801956>> (27 July 2011) 97 at 98-99; Sarah H Nash and Samuel T Redwine Jr, “People and Organizations in Software Production: A Review of the Literature” (1988) 11(3) *ACM SIGCPR Computer Personnel* 10; Steve McConnell, “10x Software Development” <<http://blogs.construx.com/blogs/stevemcc/archive/2008/03/27/productivity-variations-among-software-developers-and-teams-the-origin-of-quot-10x-quot.aspx>> (8 June 2011).

¹³⁴ Meurig Beynon, Russell Boyatt and Zhan En Chan, “Intuition in Software Development Revisited” in Jim Buckley, John Rooksby and Roman Bednarik (eds) *Proceedings of 20th Annual Psychology of Programming Interest Group Conference* (2008) <<http://www.cs.st-andrews.ac.uk/~jr/papers/ppig08Proceedings.pdf>> (16 June 2011), 95 at 96.

¹³⁵ Meurig Beynon, Russell Boyatt and Zhan En Chan, “Intuition in Software Development Revisited” in Jim Buckley, John Rooksby and Roman Bednarik (eds) *Proceedings of 20th Annual Psychology of Programming Interest Group Conference* (2008) <<http://www.cs.st-andrews.ac.uk/~jr/papers/ppig08Proceedings.pdf>> (16 June 2011), 95 at 96.

¹³⁶ Meurig Beynon, Russell Boyatt and Zhan En Chan, “Intuition in Software Development Revisited” in Jim Buckley, John Rooksby and Roman Bednarik (eds) *Proceedings of 20th Annual Psychology of Programming Interest Group Conference* (2008) <<http://www.cs.st-andrews.ac.uk/~jr/papers/ppig08Proceedings.pdf>> (16 June 2011), 95 at 96.

¹³⁷ A number of various approaches, and their problems, are canvassed in Daniel M Berry, “The Inevitable Pain of Software Development: Why There Is No Silver Bullet” in Martin Wirsing, Alexander Knapp and Simonetta Balsamo (eds), *Radical Innovations of Software and Systems Engineering in the Future, Proceedings of the 2002 Monterey Conference*, LNCS 2941 (Springer, 2004) 50.

¹³⁸ “A little retrospection shows that although many fine, useful software systems have been designed by committees and built as part of multipart projects, those software systems that have excited passionate fans are those that are the products of one or a few designing minds, great designers.”: Frederick P Brooks Jr, “No Silver Bullet: Essence and Accidents of Software Engineering” (1987) 20(4) *Computer* 10 at 18.

Is programming a fine art or a useful art?

When productivity is determined by human factors, it is suggested that aesthetic perceptions overbears rational analysis.¹³⁹ It is possible to set out three ways in which the aesthetic influences underly software development, being those set out by Sinclair in relation to mathematics, and discussed in the Chapter 5 – namely the evaluative, motivational and generative.

Evaluative aesthetic

It was noted above that when implementing a particular algorithm, it is possible to assess the efficiency of code by the extent to which it uses the physical resources of the computer on which it executes. Code that requires less CPU cycles to execute would be more efficient than code which does not. But this is not the only criterion by which code might be assessed. The following discussion summarises the possibilities:

Different programmers will have different definitions for great code. Therefore, it is impossible to provide an all-encompassing definition that will satisfy everyone. However, there are certain attributes of great code that nearly everyone will agree on, and we'll use some of these common characteristics to form our definition. For our purposes, here are some attributes of great code:

- Great code uses the CPU efficiently (that is, the code is fast).
- Great code uses memory efficiently (that is, the code is small).
- Great code uses system resources efficiently.
- Great code is easy to read and maintain.
- Great code follows a consistent set of style guidelines.
- Great code uses an explicit design that follows established software engineering conventions.
- Great code is easy to enhance.
- Great code is well tested and robust (that is, it works).
- Great code is well documented.

We could easily add dozens of items to this list. Some programmers, for example, may feel that great code must be portable, must follow a given set of programming style guidelines, must be written in a certain language or must *not* be written in a certain language. Some may feel that great code must be written as simply as possible while others may feel that great code is written quickly. Still others may feel that great code is created on time and under budget. And you can think of additional characteristics.¹⁴⁰

¹³⁹ "In programming, as in many fields, the hard part isn't solving problems, but deciding what problems to solve. Imagination is hard to measure, but in practice it dominates the kind of productivity that's measured in lines of code.": Paul Graham "Great Hackers" <<http://www.paulgraham.com/gh.html>> (16 June 2011).

¹⁴⁰ Randall Hyde, *Write Great Code: Understand the Machine* (No Starch Press, 2004) at 8-9.

Chapter 7: Why Programming is not among the Useful Arts

Some of these attributes clearly contemplate efficiency of use of the physical resources provided by the machine as a goal. These are in the nature of rational, engineering criteria. However other attributes suggest more aesthetic considerations, such as how easy such code is to read and maintain, and its consistency with the social norms which a programming team has adopted. Perhaps the most telling feature however is the admission at the beginning of the quoted passage that different programmers are likely to have different definitions. Code which is efficient may nonetheless be considered “bad” because it is difficult to extend, hard to maintain, or impossible to read.

This is a product of the fact that “you can reach the same goal ... by coding in many different styles, using different modules and deploying the same modules in different ways.”¹⁴¹ The subjective nature of the attributes of great code is indicative of the aesthetic nature of *all* of these criteria, since beauty is, after all, in the eye of the beholder. Sinclair notes in relation to mathematics, the aesthetic fulfils an evaluative role because:

mathematical reality cannot provide its own criteria; that is, a mathematical result cannot be judged important because it matches some supposed mathematical reality—mathematics is not self-organized.¹⁴²

The same is true of software development. Even seemingly rational criteria like the efficient use of resources are arbitrary. As noted above in relation to optimisation, a focus on efficiency may impact negatively on software development, since it adversely effects other criteria such as readability, complexity and may thereby reduce maintainability.¹⁴³

It has been suggested that aesthetic considerations are also “actively involved in mathematicians’ decisions about expressing and communicating their own work”.¹⁴⁴ This is clearly also the case in software, where readability, or the related criterion of maintainability is suggested as an important quality of good code.¹⁴⁵

Motivational aesthetic

The motivational aesthetic encompasses the motivations which guide the actions of programmers when writing software. As with mathematics, when it comes to programming, it is “difficult to argue that there is an objective perspective – a ... reality against which the value of

¹⁴¹ Stas Bekman and Erik Cholet, *Practical mod_perl* (2003) at 453. As the authors note, this fact is celebrated by the “main motto” of the Perl programming language, the acronym TMTOWDI (there’s more than one way to do it).

¹⁴² Nathalie Sinclair, “The Roles of the Aesthetic in Mathematical Inquiry” (2004) 6(3) *Mathematical Thinking and Learning* 261 at 265.

¹⁴³ See the discussion on page 280 above.

¹⁴⁴ Nathalie Sinclair, “The Roles of the Aesthetic in Mathematical Inquiry” (2004) 6(3) *Mathematical Thinking and Learning* 261 at 266.

¹⁴⁵ “Programs must be written for people to read, and only incidentally for machines to execute.”: Harold Abelson and Gerald Jay Sussman, *Structure and Interpretation of Computer Programs* (2nd ed, MIT Press, 1996) at xvii. The approach known as literate programming favours readability over all other considerations. See Donald E Knuth, *Literate programming* (Center for the Study of Language and Information, Stanford University, 1992) at 99: “Let us change our traditional attitude to the construction of programs: Instead of imagining that our main task is to instruct a computer what to do, let us concentrate rather on explaining to human beings what we want a computer to do.”

[software] products can be measured. Contrast this with physics, for example, another discipline that makes strong aesthetic claims ... where questions and products can be measured up against physical reality: How well they explain the shape of the universe or the behaviour of light."¹⁴⁶ The lack of objective criteria against which to make value judgements about software was discussed in relation to the evaluative aesthetic above. Beyond that, the motivational aesthetic concerns notions like the way that an aesthetic stimuli¹⁴⁷ can motivate programmers, either as an indication of the fruitfulness of an avenue of research, affirming the interest in and value of a problem area, and even sustaining a persistent pursuit over a number of years.¹⁴⁸

At a general level, the motivational aesthetic is to be found in the claims that programming is "fun". Anecdotal evidence supports the claim that software developers, or at least the best software developers, are driven by aesthetic factors, rather than strictly logical ones:¹⁴⁹

[t]heir defining quality is probably that they really love to program. Ordinary hackers write code to pay the bills. Great hackers think of it as something they do for fun, and which they're delighted to find that people will pay them for.¹⁵⁰

Given the empirical evidence which suggests a variance in programmer productivity of an order of magnitude,¹⁵¹ the importance of the motivational aesthetic is greater than it might otherwise seem. So what makes programming fun? Fred Brooks in *The Mythical Man Month* gives a detailed account of what makes programming fun, which is set out in detail below:

Why is programming fun? What delights may its practitioner expect as his reward?

First is the sheer joy of making things. As the child delights in his mud pie, so the adult enjoys building things, especially things of his own design. I think this delight must be an image of God's delight in making things, a delight shown in the distinctness and newness of each leaf and each snowflake.

Second is the pleasure of making things that are useful to other people. Deep within, we want others to use our work and to find it helpful. In this respect the programming system is not essentially different from the child's first clay pencil holder "for Daddy's office."

¹⁴⁶ Nathalie Sinclair, "The Roles of the Aesthetic in Mathematical Inquiry" (2004) 6(3) *Mathematical Thinking and Learning* 261 at 274.

¹⁴⁷ Sinclair lists the following in relation to mathematics: visual appeal, simplicity, order, "fit", surprise, paradox and the social dimension as evoking a motivational response. See Nathalie Sinclair, "The Roles of the Aesthetic in Mathematical Inquiry" (2004) 6(3) *Mathematical Thinking and Learning* 261 at 275-277.

¹⁴⁸ See Nathalie Sinclair, "The Roles of the Aesthetic in Mathematical Inquiry" (2004) 6(3) *Mathematical Thinking and Learning* 261 at 274- 275.

¹⁴⁹ In relation to the Linux kernel, the creator and project leader, Linus Torvalds claims "[t]he most important design issue ... is that Linux is supposed to be fun": in Ann Brashares, *Linus Torvalds, Software Rebel* (Twenty-First Century Books, 2001) at 45. Open source software pioneer Eric S Raymond similarly notes in respect of open source software, "[w]e have fun doing what we do ... Our creative play has been racking up technical, market share, and mind-share success at an astounding rate. We're proving not only that we can do better software, but that joy is an asset.": Eric S Raymond, *The Cathedral and the Bazaar: musings on Linux and Open Source by an accidental revolutionary* (O'Reilly Media, 2001) at 60. Python language creator Guido van Rossum similarly notes "[i]f there were no art in it, it wouldn't be any fun, and then I wouldn't still be doing it after 30 years.": in John Littler, "Art and Computer Programming" *O'Reilly OnLamp.com*, 30 June 2005 <<http://onlamp.com/pub/a/onlamp/2005/06/30/artofprog.html>> (25 May 2011).

¹⁵⁰ Paul Graham "Great Hackers" <<http://www.paulgraham.com/gh.html>> (16 June 2011).

¹⁵¹ See above n 133 on page 296 and accompanying text.

Chapter 7: Why Programming is not among the Useful Arts

Third is the fascination of fashioning complex puzzle-like objects of interlocking moving parts and watching them work in subtle cycles, playing out the consequences of principles built in from the beginning. The programmed computer has all the fascination of the pinball machine or the jukebox mechanism, carried to the ultimate.

Fourth is the joy of always learning, which springs from the nonrepeating nature of the task. In one way or another the problem is ever new, and its solver learns something: sometimes practical, sometimes theoretical, and sometimes both.

Finally, there is the delight of working in such a tractable medium. The programmer, like the poet, works only slightly removed from pure thought-stuff. He builds his castles in the air, from air, creating by exertion of the imagination. Few media of creation are so flexible, so easy to polish and rework, so readily capable of realizing grand conceptual structures...

Yet the program construct, unlike the poet's words, is real in the sense that it moves and works, producing visible outputs separately from the construct itself. It prints results, draws pictures, produces sounds, moves arms. The magic of myth and legend has come true in our time. One types the correct incantation on a keyboard, and a display screen comes to life, showing things that never were nor could be.

Programming then is fun because it gratifies creative longings built deep within us *and delights sensibilities we have in common with all men*.¹⁵²

The love of making is something that unites software development with traditional artistic pursuits.¹⁵³ In addition, delighting the sensibilities as a motivation adverts to the importance of aesthetics in software development.¹⁵⁴

The evaluative aesthetic also plays a role in influencing programmers to select the area they work in, and the projects they work on.¹⁵⁵ Within the scope of a particular project, choices also

¹⁵² Frederick P Brooks Jr, *The Mythical Man Month: Essays on Software Engineering* (Anniversary ed, Addison-Wesley, 2002) at 7-8 (emphasis added).

¹⁵³ "What hackers and painters have in common is that they're both makers. Along with composers, architects, and writers, what hackers and painters are trying to do is make good things. They're not doing research per se, though if in the course of trying to make good things they discover some new technique, so much the better.": Paul Graham "Hackers and Painters" <<http://www.paulgraham.com/hp.html>> (1 August 2011).

¹⁵⁴ Hunt and Thomas describe the source of such an emotional response along similar lines to Brooks: "We can create awe-inspiring works with little more than the exertion of the imagination. Why do we do it? We do it for the pleasure of watching them show it off to others, of watching them use in novel ways we'd never imagined. For the thrill of watching millions on millions of dollars in transactions flow through your application, confident in the results. For the joy of building and being part of a team, and for the satisfaction of knowing that you started with a blank canvas and produced a work of art.": Andrew Hunt and David Thomas, "The Art in Computer Programming" September 2001 <<http://media.pragprog.com/articles/other-published-articles/ArtInProgramming.pdf>> (1 August 2011) at 8. See also Donald E Knuth, "Computer Programming as an Art" (1974) 17(12) *Communications of the ACM* 667 at 670: "[P]rogramming can give us both intellectual and emotional satisfaction, because it is a real achievement to master complexity and to establish a system of consistent rules."

¹⁵⁵ "Along with good tools, hackers want interesting projects. ... [A]ny application can be interesting if it poses novel technical challenges. ... When Google was founded, the conventional wisdom among the so-called portals was that search was boring and unimportant. But the guys at Google didn't think search was boring, and that's why they do it so well.": Paul Graham "Great Hackers" <<http://www.paulgraham.com/gh.html>> (16 June 2011).

need to be made about the work system they use.¹⁵⁶ Programmers, for whom the computer is the whole of their work environment can be inspired by the aesthetics of the tools they have to work with – text editors, languages, version control systems, libraries and frameworks.¹⁵⁷ Even visual components such as typography may have a role to play.¹⁵⁸ Tool selection may also be more of a social, than a technological choice.¹⁵⁹ Evidence of the motivational aesthetic is also to be found in the impact of other factors such as the workplace,¹⁶⁰ and social factors

¹⁵⁶ A number of different methodologies are available, each with “a way to manage complexity and change so as to delay and moderate the [time where the software’s structure has so decayed that it is very difficult to change anything without adding more errors than have been fixed by the change]. However, each method has a catch, a fatal flaw, at least one step that is a real pain to do, that people put off.”: Daniel M Berry, “The Inevitable Pain of Software Development: Why There Is No Silver Bullet” in Martin Wirsing, Alexander Knapp and Simonetta Balsamo (eds), *Radical Innovations of Software and Systems Engineering in the Future, Proceedings of the 2002 Monterey Conference*, LNCS 2941 (Springer, 2004) 50 at 56. Berry lists the following variants: the waterfall model, structured programming, requirements engineering, extreme programming, program generation, rapid prototyping and formal methods. Within those variants, there are further sub-variants.

¹⁵⁷ “What do great hackers want? Like all craftsmen, hackers like good tools. In fact, that’s an understatement. Good hackers find it unbearable to use bad tools. They’ll simply refuse to work on projects with the wrong infrastructure.”: Paul Graham “Great Hackers” <<http://www.paulgraham.com/gh.html>> (16 June 2011). See also Donald E Knuth, “Computer Programming as an Art” (1974) 17(12) *Communications of the ACM* 667 at 672: “[I]t is still a pleasure to do routine jobs if we have beautiful things to work with. ... Please give us tools that encourage us to write better programs, by enhancing our pleasure when we do so.” Bruce J MacLennan, “Aesthetics in Software Engineering” (Technical Report No UT-CS-06-579, Department of Electrical Engineering and Computer Science, University of Tennessee, 31 October 2006) <<http://web.eecs.utk.edu/~library/TechReports/2006/ut-cs-06-579.pdf>> (26 May 2011) at 5, makes relevant observations drawn more generally to all those who use computers, drawing a useful analogy between architecture and software development:

[F]or many people the computer is not simply one tool in an otherwise uncomputerized occupation; rather, the computer and its software constitute, to a large degree, the entire occupation. In these cases the software system defines the work environment as fundamentally as the physical workspace does. Therefore, the aesthetics of the software systems deserves at least as much attention as that due the architecture, decor, etc. (From this perspective, many contemporary programs are the software equivalent of sweatshops: cluttered, dangerous, ugly, alienating, and dehumanizing.) As architecture deals with the functionality and aesthetics of physical space, organizing it for practicality and beauty, so software engineers organize cognitive (or virtual) space toward the same ends. Thus software aesthetics can have a major effect on quality of work and quality of life.

¹⁵⁸ Philip L Frana, “An Interview with Donald E Knuth”, *University of Minnesota Digital Conservancy*, 8 November 2001, <<http://conservancy.umn.edu/bitstream/107413/1/oh332dk.pdf>> (1 August 2011) at 17-18: “[O]ne of the whole ideas of structured and literate programming is that you have to be able to understand [the program’s] complicated whole. ... With good typography you can perceive the structure, instead of imagining the text as just a chaotic string of characters.”

¹⁵⁹ “When you decide what infrastructure to use for a project, you’re not just making a technical decision. You’re also making a social decision, and this may be the more important of the two. ... [W]hen you choose a language, you’re also choosing a community.”: Paul Graham “Great Hackers” <<http://www.paulgraham.com/gh.html>> (16 June 2011). “Aesthetic appreciation can unite a software development organization through a common set of values embodied in a shared sense of elegance.”: Bruce J MacLennan, “Aesthetics in Software Engineering” (Technical Report No UT-CS-06-579, Department of Electrical Engineering and Computer Science, University of Tennessee, 31 October 2006) <<http://web.eecs.utk.edu/~library/TechReports/2006/ut-cs-06-579.pdf>> (26 May 2011) at 4.

¹⁶⁰ “After software, the most important tool to a hacker is probably his office. Big companies think the function of office space is to express rank. But hackers use their offices for more than that: they use their office as a place to think in. And if you’re a technology company, their thoughts are your product.”: Paul Graham “Great Hackers” <<http://www.paulgraham.com/gh.html>> (16 June 2011). An empirical study by DeMarco and Lister backs up this claim, suggesting the work environment impacts significantly on programmer productivity. See Tony DeMarco and Tim Lister, “Programmer Performance and the Effects of the Workplace” in *Proceedings of the Eighth International Conference on Software Engineering*, Longon, August 1985, 268. See also Sarah H Nash and Samuel T Redwine Jr, “People and Organizations in Software Production: A Review of the Literature” (1988) 11(3) *ACM SIGCPR Computer Personnel* 10 at 14: “Motivation is correlated with productivity and keyed to the work environment.”

Chapter 7: Why Programming is not among the Useful Arts

such as team dynamics.¹⁶¹

Even with these preliminary and external factors in place, there are design decisions to be made about algorithms¹⁶² and data structures to use, and how to express them. As in mathematics, the aesthetic may guide these selections. Given the highly complex nature of software creations, rational analysis of the impact of change on a system becomes impossible.¹⁶³ Therefore software developers are reliant on aesthetic judgments, simply put, that “designs that look good are good”.¹⁶⁴

Generative aesthetic

Given the personal, and largely internal nature of much of the activity of programming, there is little direct evidence of the way in which the aesthetic influences the process of inquiry. Undoubtedly this is because we don’t understand the way in which the spark of creativity ignites, generally, since it operates at a “tacit or even subconscious level, and intertwined as it frequently is with intuitive modes.”¹⁶⁵ But it is suggested that the role of the aesthetic in “the discovery and invention of solutions or ideas; in guiding the actions and choices that [software developers] make as they try to make sense of objects and relations.”¹⁶⁶

In the context of mathematics, Sinclair identifies three strategies mathematicians use for evoking the generative aesthetic: “playing, establishing intimacy, and capitalizing on intuition”.¹⁶⁷

Play Sinclair notes the role of “free play” in allowing mathematicians to develop their craft. “[T]he mathematician, freed from having to solve a specific problem using the analytical apparatus of her craft, can focus on looking for appealing structures, patterns and combinations

¹⁶¹ See the review of relevant studies on this point in Sarah H Nash and Samuel T Redwine Jr, “People and Organizations in Software Production: A Review of the Literature” (1988) 11(3) *ACM SIGCPR Computer Personnel* 10, at 14-15. Brooks also acknowledges this social aspect in his discussion of the ways of developing the next generation of top software designers, when he stresses mentoring, apprenticeships with top designers, and providing opportunities for people to “interact with and stimulate each other”. See Frederick P Brooks Jr, “No Silver Bullet: Essence and Accidents of Software Engineering” (1987) 20(4) *Computer* 10 at 18.

¹⁶² “Given a solvable problem, there are many algorithms (programs) to solve it, not all of equal quality. The primary practical criteria by which the quality of an algorithm is judged are time and memory requirements, accuracy of solution, and generality.”: *Encyclopedia of Computer Science*, (Wiley & Sons, 2003) <<http://www.credoreference.com/entry/5880599>> (27 July 2011). It should be noted however, despite the seemingly technical choice just described, that algorithms are not often chosen like off-the-shelf components. In many instances, the algorithm eventually emerges from the repeated attempts of the programmer to describe the method in computationally descriptive form. This is what Brooks means when he says that software is “grown, not built.”: Frederick P Brooks Jr, “No Silver Bullet: Essence and Accidents of Software Engineering” (1987) 20(4) *Computer* 10 at 18.

¹⁶³ Bruce J MacLennan, “Aesthetics in Software Engineering” (Technical Report No UT-CS-06-579, Department of Electrical Engineering and Computer Science, University of Tennessee, 31 October 2006) <<http://web.eecs.utk.edu/~library/TechReports/2006/ut-cs-06-579.pdf>> (26 May 2011) at 5.

¹⁶⁴ Bruce J MacLennan, “Aesthetics in Software Engineering” (Technical Report No UT-CS-06-579, Department of Electrical Engineering and Computer Science, University of Tennessee, 31 October 2006) <<http://web.eecs.utk.edu/~library/TechReports/2006/ut-cs-06-579.pdf>> (26 May 2011) at 3-4.

¹⁶⁵ Nathalie Sinclair, “The Roles of the Aesthetic in Mathematical Inquiry” (2004) 6(3) *Mathematical Thinking and Learning* 261 at 270.

¹⁶⁶ Nathalie Sinclair, “The Roles of the Aesthetic in Mathematical Inquiry” (2004) 6(3) *Mathematical Thinking and Learning* 261 at 270.

¹⁶⁷ Nathalie Sinclair, “The Roles of the Aesthetic in Mathematical Inquiry” (2004) 6(3) *Mathematical Thinking and Learning* 261 at 271.

of ideas.”¹⁶⁸ Some programmers suggest a similar ploy for developing proficiency, through the writing of “toy programs”.¹⁶⁹ The camel code in Listing ?? above is a good example of such a program. On a similar note, programmers involve themselves in open source software development projects beyond paid work in order to develop their proficiency. Other forms of experimentation include participation in obfuscated code contests, and the development of “weird” languages.¹⁷⁰

In relation to particular projects, the early stages of a project may involve free play in a different form, by simply doodling on paper, exploring structures and concepts in order to come to an understanding of the nature of the project domain. Some approaches to software development, such as rapid prototyping and incremental release methodologies such as agile and XP, also have this sort of aesthetic experimentation at their core, since they encourage building code as a way of exploring a domain, and generating aesthetic feedback through use, or experience, of the software from an early stage. More formal approaches such as requirements engineering, and structured programming, use other written works such as formal specifications as the mechanism of exploration.

Personal, intimate relationships Sinclair also draws attention to the way mathematicians become familiar with an area by naming the objects they are dealing with, with a view to getting “some traction on still vague territory”.¹⁷¹ There is some evidence of this sort of anthropomorphisation by programmers.¹⁷² However, the development of a relationship between programmer and code is centred around the development of a prototype. This desire to develop the coder-code relationship explains the “code first, then debug” approach adopted by many novices, and even some experienced programmers.¹⁷³ Developing a relationship with the domain is also at the heart of more modern iterative approaches such as spiral development,

¹⁶⁸ Nathalie Sinclair, “The Roles of the Aesthetic in Mathematical Inquiry” (2004) 6(3) *Mathematical Thinking and Learning* 261 at 272.

¹⁶⁹ Knuth - Art, 672. Dave Thomas, advocates a similar approach, which he calls “Code Kata”: http://codekata.pragprog.com/codekata/2007/01/code_katahow_it.html (11 June 2011).

¹⁷⁰ See Thomas Taylor, “Obfuscation, Weird Languages and Code Aesthetics” on *obfuscators.org*, 26 April 2008 <http://www.obfuscators.org/2008/04/obfuscation-weird-languages-and-code.html> (1 August 2011).

¹⁷¹ Nathalie Sinclair, “The Roles of the Aesthetic in Mathematical Inquiry” (2004) 6(3) *Mathematical Thinking and Learning* 261 at 272.

¹⁷² See for example Eric S Raymond, “Anthropomorphization” in *The Online Hacker Jargon File* <http://www.catb.org/~esr/jargon/html/anthropomorphization.html>. As Raymond notes, this anthropomorphization extends to both software and hardware.

¹⁷³ See Daniel M Berry, “The Inevitable Pain of Software Development: Why There Is No Silver Bullet” in Martin Wirsing, Alexander Knapp and Simonetta Balsamo (eds), *Radical Innovations of Software and Systems Engineering in the Future, Proceedings of the 2002 Monterey Conference*, LNCS 2941 (Springer, 2004) 50 at 51-53; Boehm discusses the evolution of the hacker culture, which lauds such cowboy programmers in Barry Boehm, “A View of 20th and 21st Century Software Engineering” (Paper presented at the 28th International Conference on Software Engineering, Shanghai, China, 20 May 2006) 12 <http://portal.acm.org/citation.cfm?id=1134288> at 13; Paul Graham asserts a personal view that this is the only way to develop software:

I was taught in college that one ought to figure out a program completely on paper before even going near a computer. I found that I did not program this way. I found that I liked to program sitting in front of a computer, not a piece of paper. Worse still, instead of patiently writing out a complete program and assuring myself it was correct, I tended to just spew out code that was hopelessly broken, and gradually beat it into shape. Debugging, I was taught, was a kind of final pass where you caught typos and oversights. The way I worked, it seemed like programming consisted of debugging. ... [T]here was a name for what I was doing: sketching. As far as I can tell, the way they taught me to program in college was all wrong. You should figure out programs as you're writing them, just as writers and painters and architects do.”

Chapter 7: Why Programming is not among the Useful Arts

agile, extreme programming.¹⁷⁴ Although other more formal, structured methodologies don't involve an early relationship with code in a strict sense, requirements documentation, formal analysis, requirements engineering, are geared towards building familiarity with the concepts that lie behind the code, and may therefore be treated as involving a similar process.¹⁷⁵

The need for a relationship with the code is also behind one of the great drivers of change in software projects, the "IKIWISI" phenomenon,¹⁷⁶ whereby the early prototype informs not only programmers' conception of the software being created, but also the client or end user's.¹⁷⁷ This in part explains the drive to change software throughout its development.

Finally Brooks' notion that software should be grown, not built, is a reflection of "an organic process of interaction between the embryonic software product and its environment that takes place in the developer's mind."¹⁷⁸

Capitalising on intuition Peter Naur, an important figure in computer science, advanced a position in the 1984 that intuition was "the basis on which all activities involved in software development must build."¹⁷⁹ Despite Naur's status as a "most distinguished contributor to the study of software practice, and ... a recipient of the Turing Award in 2005",¹⁸⁰ the paper has received only a handful of references in subsequent academic literature.¹⁸¹ Adherents to a strict, formalistic, or rationalist approach to software development are likely to equate intuition with guesswork, or a lack of risk aversion.¹⁸² So is it relevant to suggest a central role for intuition in mathematics?

Paul Graham "Hackers and Painters" <<http://www.paulgraham.com/hp.html>> (1 August 2011).

¹⁷⁴ These methodologies, and their weaknesses are discussed in Daniel M Berry, "The Inevitable Pain of Software Development: Why There Is No Silver Bullet" in Martin Wirsing, Alexander Knapp and Simonetta Balsamo (eds), *Radical Innovations of Software and Systems Engineering in the Future, Proceedings of the 2002 Monterey Conference*, LNCS 2941 (Springer, 2004) 50 at 58-65.

¹⁷⁵ See Daniel M Berry, "The Inevitable Pain of Software Development: Why There Is No Silver Bullet" in Martin Wirsing, Alexander Knapp and Simonetta Balsamo (eds), *Radical Innovations of Software and Systems Engineering in the Future, Proceedings of the 2002 Monterey Conference*, LNCS 2941 (Springer, 2004) 50 at 58-61.

¹⁷⁶ This stands for "I'll know it when I see it".

¹⁷⁷ In this author's experience, and humble opinion, it often seems to be the case that clients don't really have any conception of what they want, but are quite adept at explaining how a prototype is *not* what they want. (refs?)

¹⁷⁸ Meurig Beynon, Russell Boyatt and Zhan En Chan, "Intuition in Software Development Revisited" in Jim Buckley, John Rooksby and Roman Bednarik (eds) *Proceedings of 20th Annual Psychology of Programming Interest Group Conference* (2008) <<http://www.cs.st-andrews.ac.uk/~jr/papers/ppig08Proceedings.pdf>> (16 June 2011) at 99.

¹⁷⁹ Peter Naur, "Intuition in Software Development" in Harmut Ehrig et al (eds) *Theory and Practice of Software Development*, Volume 2, LNCS 186 (Springer, 1985) 60. For a more recent update, see Meurig Beynon, Russell Boyatt and Zhan En Chan, "Intuition in Software Development Revisited" in Jim Buckley, John Rooksby and Roman Bednarik (eds) *Proceedings of 20th Annual Psychology of Programming Interest Group Conference* (2008) <<http://www.cs.st-andrews.ac.uk/~jr/papers/ppig08Proceedings.pdf>> (16 June 2011).

¹⁸⁰ Meurig Beynon, Russell Boyatt and Zhan En Chan, "Intuition in Software Development Revisited" in Jim Buckley, John Rooksby and Roman Bednarik (eds) *Proceedings of 20th Annual Psychology of Programming Interest Group Conference* (2008) <<http://www.cs.st-andrews.ac.uk/~jr/papers/ppig08Proceedings.pdf>> (16 June 2011) at 95.

¹⁸¹ Meurig Beynon, Russell Boyatt and Zhan En Chan, "Intuition in Software Development Revisited" in Jim Buckley, John Rooksby and Roman Bednarik (eds) *Proceedings of 20th Annual Psychology of Programming Interest Group Conference* (2008) <<http://www.cs.st-andrews.ac.uk/~jr/papers/ppig08Proceedings.pdf>> (16 June 2011) at 95.

¹⁸² "One of the main purposes of adopting a formal approach to program development is to shift the focus from 'considering your intuitive ideas about what a program is intended to do' to 'thinking precisely and abstractly about what is it [sic] that you have instructed it to do'." Meurig Beynon, Russell Boyatt and Zhan En Chan, "Intuition in Software Development Revisited" in Jim Buckley, John Rooksby and Roman Bednarik (eds) *Proceedings of 20th Annual Psychology of Programming Interest Group Conference* (2008) <<http://www.cs.st-andrews.ac.uk/~jr/papers/ppig08Proceedings.pdf>> (16 June 2011) at 96 (citations omitted).

Poincaré characterised the mathematical generative process in mathematics as being concerned with, not the logical, but “the construction of possible combinations of ideas and the selection of the fruitful ones.”¹⁸³ As a matter of intuition, experience,¹⁸⁴ and on the basis of the Curry-Howard isomorphism, it is asserted that software development is similarly concerned with the combination and selection of such ideas. The aesthetic assists in the assessment of combinations of ideas which are “harmoniously disposed so that the mind can effortlessly embrace their totality without realising their details.”¹⁸⁵ The intangible nature of software¹⁸⁶ means that the mind’s traditional tools for dealing with spatio-temporal objects are not available.¹⁸⁷ The complexity of software creations means that it is very difficult to rationally assess the effect of software design decisions or changes on the whole of software.¹⁸⁸ Ershov aptly captures the complexity of the programming task, which he describes as “the most humanly difficult of professions”,¹⁸⁹ because:

In his work, the programmer is challenged to combine, with the ability of a first-class mathematician to deal in logical abstractions, a more practical, a more Edisonian talent, enabling him to build useful engines out of zeros and ones, alone. He must join the accuracy of a bank clerk with the acumen of a scout, and to these add the powers of fantasy of an author of detective stories and the sober practicality of a businessman. To top all this off, he must have a taste for collective work and a feeling for the corporate interests of his employer.¹⁹⁰

In other words, “[t]he scale of software development is such that no single designer can appreciate all the perspectives that are relevant to the effective solution of problems.”¹⁹¹ Therefore

¹⁸³ Nathalie Sinclair, “The Roles of the Aesthetic in Mathematical Inquiry” (2004) 6(3) *Mathematical Thinking and Learning* 261 at 270.

¹⁸⁴ See also the comments of Charles Strauss in n 115 on page 292 above.

¹⁸⁵ Nathalie Sinclair, “The Roles of the Aesthetic in Mathematical Inquiry” (2004) 6(3) *Mathematical Thinking and Learning* 261 at 270.

¹⁸⁶ “The essence of a software entity is a construct of interlocking concepts: data sets, relationships among data items, algorithms, and invocations of functions”: Frederick P Brooks Jr, “No Silver Bullet: Essence and Accidents of Software Engineering” (1987) 20(4) *Computer* 10 at 11.

¹⁸⁷ Frederick P Brooks Jr, “No Silver Bullet: Essence and Accidents of Software Engineering” (1987) 20(4) *Computer* 10 at 12.

¹⁸⁸ “In most cases, the elements interact with each other in some nonlinear fashion, and the complexity of the whole increases much more than linearly.”: Frederick P Brooks Jr, “No Silver Bullet: Essence and Accidents of Software Engineering” (1987) 20(4) *Computer* 10 at 11.

¹⁸⁹ Andrei P Ershov, “Aesthetics and the Human Factor in Programming” (1972) 15(7) *Communications of the ACM* 501. See also Brad Cox, “No Silver Bullet Revisited” *American Programmer Journal*, November 1995 <<http://virtualschool.edu/cox/pub/NoSilverBulletRevisited>> (22 July 2011): “[P]rogrammers invariably agree that computer software is the most complicated of all human activities. I’ve been heard [sic] word processors seriously compared with the complexity of Boeing’s airplanes.”

¹⁹⁰ Andrei P Ershov, “Aesthetics and the Human Factor in Programming” (1972) 15(7) *Communications of the ACM* 501 at 502. Ershov lists a further two reasons why software creation is such a complex task, namely that their work “brings them to those limits of human knowledge which are marked by algorithmically unsolvable problems and which touch upon deeply secret aspects of the human brain”: *ibid*, and that the programmer’s personal push-down stack “must be as deep as is needed for the problem, plus at least 2-3 positions deeper”: *ibid*. A personal push-down stack is a reference to the ability to juggle tasks in a last-in-first-out fashion. He asserts that the average person has a stack depth of 5-6 tasks. Ershov clarifies what he means by such a stack in a contemporaneous letter to an enquirer. Letter from Andrei P Ershov to Frederic L Coombes, 18 November 1972 <<http://ershov.iis.nsk.su/archive/eaindex.asp?lang=2&did=382>> (21 June 2011).

¹⁹¹ Meurig Beynon, Russell Boyatt and Zhan En Chan, “Intuition in Software Development Revisited” in Jim Buckley, John Rooksby and Roman Bednarik (eds) *Proceedings of 20th Annual Psychology of Programming Interest Group Conference* (2008) <<http://www.cs.st-andrews.ac.uk/~jr/papers/ppig08Proceedings.pdf>> (16 June 2011) at 98.

Chapter 7: Why Programming is not among the Useful Arts

it is only through aesthetic impressions that the software design and construction process is informed during the generative phase. Indeed it may be that Brooks' formulation of "conceptual integrity" as a paramount consideration in successful software projects is referable to the intuitive understanding of the project in the mind of the project leader.¹⁹² A recognition of the aesthetic in programming demands "complete recognition and full exploitation of the broad scale of individual activities in programming."¹⁹³

Summary

The foregoing discussion underscores the creative nature of the programmer. Aesthetic considerations influence every aspect of software development, including: assisting in the evaluation of code and infrastructure quality; guiding project selection and influencing productivity; and guiding the activities of individual developers. The aesthetic also features in the way that programmers use experimentation and intuition to manage complexity, perceive solutions, and advance the state of the art. Software development does not proceed by a sequence of logical steps, but, as with any creative endeavour, involves a winding, internal and seemingly irrational path from concept to code.

Programming is not in the useful arts

To review, the results of the 3-dimensional analysis are as follows. Firstly, it is entirely possible to analyse claims to software on a physical level. On such a view, the physicality of software has been located by courts in the computer hardware on which software eventually executes, and the medium on which software is stored before execution. But there is a significant difference between software, as written, and software as executing. To focus on the end product, agonising about what software is, and to point to a particular physical device as being the locus of software's existence is to ignore the aspect of software which is most significant to its creation. As far as programmers (and software's users) are concerned, it is the symbolic aspects of software which are most important.

This being the case, its abstractness or physicality – the context of the claimed invention – needs to be very carefully considered to determine whether the physical aspects of the computer hardware, or associated peripherals, impose any meaningful limitation on the development of the software. Another way of putting it is to ask whether the software is merely a component of a larger device (not being a general purpose computer) or whether what is claimed is an abstract artefact, or a physical invention. Artefacts existing higher up the stack, such as algorithms and abstract data structures are sufficiently abstract that they ought never to be considered patentable. As noted in Chapter 1, the history of software development suggests

¹⁹² Meurig Beynon, Russell Boyatt and Zhan En Chan, "Intuition in Software Development Revisited" in Jim Buckley, John Rooksby and Roman Bednarik (eds) *Proceedings of 20th Annual Psychology of Programming Interest Group Conference* (2008) <<http://www.cs.st-andrews.ac.uk/~jr/papers/ppig08Proceedings.pdf>> (16 June 2011) at 99.

¹⁹³ Andrei P Ershov, "Aesthetics and the Human Factor in Programming" (1972) 15(7) *Communications of the ACM* 501 at 505.

Is programming a fine art or a useful art?

that, in general, software will be crafted at higher and higher levels of abstraction. Even at the present time, the majority of software is written at a level of abstractness beyond the realm of meaningful physical limitations.

Secondly, software implements a function, namely a computable process. However, the expressiveness of software remains a significant part of the software development process. It permeates every level of the abstraction stack, and is not stripped from software even by its translation to executable form. Even in that form it contains, at the very least a mere kernel of technical expressiveness, such as that which most functional devices are likely to contain. Beyond that, it is submitted that executable software is expressive in other ways. In a very real sense, software communicates to users the way in which it is to be used to achieve particular goals. The machine code itself also maintains an informative aspect, in that it continues to describe a particular computable process, which with the benefit of decompilation tools, such information can be extracted. It will only really be where the physical aspects of computer hardware significantly restricts the number of ways in which a particular process can be implemented that the expressiveness of software is curtailed. Further, as software continues to grow more and more abstract, software development is likely to become more and more expressive over time.

Finally, the foundation in logic and formal mathematics, as well as the business-like nature of the enterprise, and the desire to make software into a form of engineering, all create an impression of software development as a rational activity. But software is not created by progressing through a sequence of logical steps, rather by travelling a winding and tortuous path. Software development is a creative, artistic pursuit, governed by aesthetic considerations. Aesthetics determines the criteria by which software is adjudged good or bad, draws programmers towards particular development projects, motivates developers to create, and is key to the advancement of the discipline through the generation of innovative approaches to development. Although the development of software may be assisted by structured methodologies, they cannot address the essence of software development, only its accidents, and are therefore only of secondary importance. This will be the case irrespective of the level of physicality of the claimed invention, since in all cases, the activity of programming is directed to crossing the semantic gap from a human understanding of desired function, to a complete and precise description of a computable process.

Taking all this into account, it is clear that the creation of software, in all but exceptional circumstances, falls within the fine arts, rather than the useful arts. Undoubtedly there will be occasions where the factual matrix of a particular set of claims ought to be considered very carefully, before a final decision is made. But, as noted in the Chapter 4,¹⁹⁴ it is where the distinctions are the slightest, that the most resolute adherence to such distinctions is required. The framework just used holds value as a way of analysing a particular set of claims, to determine how physical, how expressive and how aesthetic the development process is likely to be, and whether therefore such a claimed invention is patentable subject matter. But the general analysis just completed makes clear that the starting point for any claims to software should

¹⁹⁴ See Chapter 4, at 156.

Chapter 7: Why Programming is not among the Useful Arts

be that software is *not* patentable. This being the case, a categorical exclusion for software is appropriate.

Where what is claimed involves a software component, but the creation of such software is dominated by physical, functional and rational considerations, then it might be concluded that the essence of that invention is not the software which is claimed, but the physical hardware, and that as such the invention falls outside of a categorical exclusion. Such a position is consistent with the claimed isomorphism between mathematics and software, since the “software” in such instances is a mere description of a component of a physical device, which accords with the orthodox view that a mathematical description of the parts of a physical invention would not be a bar to patentability.

3 Conclusion

This chapter has looked at the patentability of software, as determined by the characteristics of the activity involved in creating it, namely programming. There is no doubt that software falls near the borderline of patentable subject matter. But for the reasons set out above, it is clear that it falls outside the realm of patentable subject matter. When what is claimed is the product of programmers’ creative endeavours, it will in most cases be an intangible, expressive, and aesthetic artefact which is claimed, and is not within the field of the useful or technological arts.

The contention that software is not patentable subject matter has clear implications, both for the software industry, the patent regime, and intellectual property more broadly. It is to those implications that the final chapter of this thesis is devoted.

Implications

1 Introduction

The last chapter demonstrated why programming, like mathematics, is not a useful art, with the consequence that software ought not be patentable. This chapter explores the implications of that position.¹ First of all, the bounds of subject matter as presently understood in the three jurisdictions considered in this thesis are considered in light of the analytical framework adopted.

Although software development is a fine art, rather than a useful art, simply declaring that software is not patentable is unlikely to be dispositive of the issue. The history of the exclusion of computer programs “as such” under the European Patent Convention stands as a testament to that. With the growing ubiquity of computers, software components have become a part of many industrial processes, and components of all manner of common physical devices, from watches to cars. These products are part of the traditional domain of patent law. Therefore, there is at the very least a need to distinguish claims to a patentable invention containing a computable process, and non-patentable claims to the process itself. This stresses the importance of properly characterising the invention.

Next, the role of subject matter is considered. The need to exclude software from the patent paradigm suggests that subject matter is not a “failed gatekeeper” which should be retired. Some consideration is also given to the role of freedom in patent law. Whilst the relationship between copyright and free expression is much better travelled, the role of freedom in patent law is much less discussed. There has been some consideration given to the influence of human rights jurisprudence on intellectual property, and in particular patent law.

Having considered the implications of the *prima facie* non-patentability of software on the patent regime, some remarks are addressed to a broader issue. As it is suggested that soft-

¹ Parts of this chapter were originally drawn from Anton R Hughes, “Submission to the Advisory Council on Intellectual Property: Patentable Subject Matter Issues Paper” (19 September 2008) <<http://www.acip.gov.au/reviewpatentable/IP22%20Anton%20Hughes.pdf>>; and Anton R Hughes, “Avoiding the Software Patent Problem: An Alternative Fix for TRIPS Junkies” (2007) 14(1) *eLaw: Murdoch University Electronic Journal of Law* 100.

Chapter 8: Implications

ware ought not to be protected within the patent regime, what are the implications of such a proposal on the protection which software should, would, or perhaps already does receive?

Because this is the final chapter, and is followed by the conclusion, it does not have its own concluding section.

2 An assessment of the framework

At the outset of the thesis, three features of software which make it special were set out: abstractness, complexity and reuse. A few comments are now directed towards how the framework developed and applied takes account of those features.

The level of abstractness of a particular invention is clearly assessed in the physicality versus intangibility dimension of the framework. The more abstract a piece of software is, the less connection it has to the workings of the physical computer on which it runs. The abstractness of software also affects the available expressivity since, as asserted in Chapter 7, it is only at very low levels of abstraction that the physical limitations of hardware will restrain the ability to express a computable process in more than one way.

The complexity of software was explored in the third dimension of the framework, the aesthetic versus the rational, and it was noted that the complexity of software means its construction cannot depend on a rational analysis of its states. As such, the complexity of software results in a reliance on aesthetic responses, or intuition, to guide the creative process. The framework clearly addresses this aspect of software's nature.

The need for reuse is related to all three limbs of the analytical framework. It is the intangibility of software which makes its reuse simple. Aesthetics depend on and are guided by previous experience. A dependence on reuse also suggests the need for freedom *to* reuse, and as such lies at the foundation of the analytical framework. To look at it another way, a culture of reuse is a hallmark of a fine art.²

3 An assessment of the state of patent law

Given the argument put forward in Chapter 7, that software ought not be patentable, it is worthwhile to compare how the current state of subject matter in the three jurisdictions considered in this thesis fares. As was noted in Chapter 7, all jurisdictions considered are signatories to the TRIPS treaty,³ Article 27 of which requires that patents be made available in all fields of

² The importance of reuse in other fine arts such as music, literature and drama is almost so evident that it goes without saying. However, an illustration of the importance of reuse to music is made at the end of the movie *Patent Absurdity: How Software Patents Broke the System* (Directed by Luca Lucarini, 2010) <<http://patentabsurdity.com/>> (27 August 2011) at 27:45. See also Richard Stallman, "The Danger of Software Patents," Transcript of Speech Given at Cambridge University, March 2002, <<http://www.cl.cam.ac.uk/~mgk25/stallman-patents.html>> (27 August 2011): "Beethoven, as it happens, had a lot of new musical ideas but he had to use a lot of existing musical ideas in order to make recognizable music."

³ *Marrakesh Agreement Establishing the World Trade Organization*, opened for signature 15 April 1994, 1867 UNTS 3, annex 1C (*Agreement on Trade-Related Aspects of Intellectual Property Rights*) (entered into force 1 January 1995).

technology. So on that basis, the three jurisdictions should be compatible with the analytical framework, since the framework determines what is in the field of technology as against what is outside it.

The United States

The United States limitation of patent law to the useful arts is obviously consistent with the analytical framework proposed. The current understanding of that test excludes “laws of nature, physical phenomena, and abstract ideas”,⁴ and relies on the machine-or-transformation test as either a de-facto test, or at least a “useful and important clue”.⁵ The machine-or-transformation test is directed to the physicality dimension of the three dimensions used in this thesis. It was noted how the expressiveness of software can depend at least to some degree on the physicality analysis, so there is at least some indirect consideration of that dimension. This expressiveness was directly raised in the *Prater* rehearing,⁶ but then the case was decided on other grounds. This could have been characterised as an avoidance of the issue, or a downplaying of its significance.

The doctrine of mental steps⁷ was theoretically directed to both the abstract nature of the process claimed. But the failure of the doctrine was not, as argued by Coulter, that it did not discriminate between technological, rational processes which could be carried out without human intervention, and those including “peculiarly human mental activities which cannot, in principle be performed by devices”.⁸ The failure of the doctrine was that it did not consider the other two dimensions of the analysis used, which are concerned with the nature of the creative process. Similar problems may attend on the “abstract ideas” exclusion, if it is said that ideas are patentable only because they are not physical.⁹ The machine-or-transformation test is on its face directed towards only one dimension of analysis as well. As was argued in Chapter 7, the physical limitations of computer hardware might reduce the expressivity of software, but say nothing about the aesthetic versus rational nature of its creation.

The majority approach in *Bilski* is more problematic. Proper distinctions between the useful arts and the fine arts cannot be made by asking whether something is a process within the “ordinary, contemporary, common meaning”¹⁰ of the word. As Stevens J correctly noted,¹¹ the supposed

⁴ *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3221 per Kennedy J.

⁵ *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3226 per Kennedy J, 3258 per Breyer J.

⁶ *In re Prater* (1969) 415 F.2d 1390, discussed in Chapter 3 at 81. The expressiveness was raised in the form of an objection that fell foul of the First Amendment right of freedom of thought.

⁷ Discussed in Chapter 4, at 2 on page 154.

⁸ Robert I Coulter, “The Field of the Statutory Useful Arts: Part I” (1952) 34 *Journal of the Patent Office Society* 425 at 426. Coulter’s approach conflates the nature of the process claimed, with the nature of the activity involved in producing it. As has been argued in this thesis, just because a process, for example, a computable process, is defined, and even carried out, as a sequence of logical steps, it does not follow that the creative process behind it is similarly logical or rational. In fact, Chapter 7 demonstrates that this is *not* the case for software.

⁹ In the recent decision in *CyberSource Corporation v Retail Decisions Inc* Appeal No 2009-1358 (Fed Cir, 2011), Dyk J held the claimed software method non-patentable as a mental process, a subcategory of the abstract ideas exception, because it “can be performed in the human mind, or by a human using a pen and paper”: at 12. Such an approach seems to reinvigorate the mental steps doctrine, in an albeit narrower way, since claims which might be performed by a human might yet be patentable if the claims are limited by a machine or transformation.

¹⁰ *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3221 per Kennedy J.

¹¹ *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3235.

Chapter 8: Implications

literal interpretation of the majority is not able to be reconciled with the specific exclusions for “laws of nature, physical phenomena and abstract ideas”.¹² In addition, the ordinary meaning of the word process says nothing about the fact that both fine and useful arts may involve processes, but it is only the latter which are traditionally patentable. To address that distinction it is necessary to consider the nature of the process by which an allegedly *patentable* process is “made”. It is urged that a full three dimensional consideration of the bounds of patentability in that jurisdiction is not precluded by *Bilski*, but would require a change in approach to interpreting the category of patentable processes, one steeped in a historical, purposive interpretation of the constitutional clause, rather than a literal one. This was the approach adopted by Stevens J, although admittedly it represents a minority view.

The exclusion of “laws of nature, physical phenomena, and abstract ideas”,¹³ does however provide a possible way out of this conundrum. As noted above, the abstract ideas exclusion might only be considered directed towards the intangible versus physical limb of the framework. However, recalling the relationship between thought and expression discussed in Chapter 6,¹⁴ such ideas are better characterised as giving rise to expressive rather than purposive considerations. As the process of creating ideas is a peculiarly human process, and one which is not well understood, such abstract ideas are better described as aesthetic than rational. The nature of mathematics, as primarily grounded in thought and expression, illustrates why the creation of such ideas is the domain of the fine arts, rather than the useful arts.

Similarly, the laws of nature exclusion can be reconciled with the analytical framework used herein. On the basis of the characterisation of mathematics as “the queen of the sciences”¹⁵ and the acknowledged aesthetics of other sciences,¹⁶ it is submitted that a distinction might be drawn between science and technology in that the former has strong aesthetic and expressive characteristics, putting it closer to the fine rather than the useful arts.

The United Kingdom

The relationship between the analytical framework and the EU model is perhaps the most challenging to clearly resolve. On the one hand, the tests used by the EPO and UKPO resolve the enumerated exclusions to a single technicality requirement (whether that be technical contribution or technical effect). Technicality could be taken to be a synonym for technology, and therefore compatible with the analytical framework. However, the resolution of the Article 52(2) categories to a requirement of technical character rests on a dubious legal foundation. Such a formulation of patentable subject matter was explicitly “rejected by the framers of the

¹² *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3221 per Kennedy J, quoting *Diamond v Chakrabarty* 447 US 303 at 309.

¹³ *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3221 per Kennedy J.

¹⁴ See Section 3 on page 237.

¹⁵ Carl Friedrich Gauss, cited in Guy W Dunnington, Jeremy Gray and Fritz-Egbert Dohse, *Carl Friedrich Gauss: titan of science* (MAA, 2004) at 44.

¹⁶ Nathalie Sinclair, “The Roles of the Aesthetic in Mathematical Inquiry” (2004) 6(3) *Mathematical Thinking and Learning* 261 at 274, describing physics as another discipline with “strong aesthetic claims”, and citing in support Graham Farmelo (ed), *It must be beautiful: Great equations of modern science*. (Granta Books, 2002).

EPC from the outset”.¹⁷ Further, as noted in *Symbian*,¹⁸ the reliance on the concept of technicality in the EPC context is problematic because Article 52 makes “no reference to any ‘technical’ requirement”.¹⁹ As such, in the absence of clear guidance as to the meaning of this concept, such a test will have “all the disadvantages of the original obscure wording, with the added disadvantage of not even providing the actual legislative test”.²⁰

It may be recalled from Chapter 3, that Article 52(2) of the European Patent Convention declares that the following are not patentable inventions “as such”.²¹

- (a) discoveries, scientific theories and mathematical methods;
- (b) aesthetic creations;
- (c) schemes, rules and methods for performing mental acts, playing games or doing business, and programs for computers;
- (d) presentations of information.

But, as noted in Chapter 3, these exclusions share a common characteristic that they are “clearly non-technical”.²² Two of those specific exclusions, namely computer programs, and mathematical methods, have been analysed according to the proposed framework. The remaining exclusions would, or a cursory analysis at least, seem consistent with that analysis as well. So to that extent, it seems fair to suggest that the analytical framework both fleshes out the features which make the specific exclusions non-technical, and may be of benefit in determining whether a particular invention falls within or without a specific exclusions.

Australia

In Australia, the original manner of manufacture test from the *Statute of Monopolies* remains in force, as interpreted in *NRDC*.²³ As noted in Chapter 2, the useful arts versus fine arts distinction was drawn in that case, and illustrates a consistent position with the US. The heavy borrowing of guidance from the US on subject matter approaches also is consistent with that position. A recent review of the patentable subject matter test by the Advisory Council on Intellectual Property recommended a change to the test, “using clear and contemporary language that embodies the principles of inherent patentability as developed by the High Court in the

¹⁷ Justine Pila, “Dispute over the Meaning of ‘Invention’ in Article 52(2) EPC – The Patentability of Computer-Implemented Inventions in Europe” (2005) 36 *International Review of Intellectual Property and Competition Law* 173 at 185.

¹⁸ *Symbian Ltd v Comptroller-General of Patents* [2008] EWCA Civ 1066; [2009] RPC 1, discussed in Chapter 3, 4 on page 136.

¹⁹ *Symbian Ltd v Comptroller-General of Patents* [2008] EWCA Civ 1066; [2009] RPC 1 at [29].

²⁰ *Symbian Ltd v Comptroller-General of Patents* [2008] EWCA Civ 1066; [2009] RPC 1 at [30].

²¹ The “as such” proviso is found in Article 52(3): “The provisions of paragraph 2 shall exclude patentability of the subject-matter or activities referred to in that provision only to the extent to which a European patent application or European patent relates to such subject-matter or activities as such.

²² Brad Sherman, “The patentability of computer-related inventions in the United Kingdom and the European Patent Office” (1991) 13(3) *European Intellectual Property Review* 85 at 93. Discussed in Chapter 3, Section on page 103.

²³ *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252, discussed in detail in Chapter 2, Section 6 on page 61.

Chapter 8: Implications

NRDC case and in subsequent Australian court decisions.”²⁴ No specific wording of that proposed codification was endorsed, although in the discussion, ACIP noted that “subject matter must relate to the useful, rather than the fine, arts”.²⁵

Chapter 2 made clear how physicality was a key component of the NRDC decision, and the most recent case of *Grant*²⁶ in Chapter 3 clearly endorsed this requirement. As is to be expected, given the way that Australian law has tracked the US position, little consideration of the other dimensions of analysis is evident. At best, it may be possible to read a consideration of expressiveness into the traditional exclusions, such as the intellectual information exclusion. As to the third dimension, aesthetics, there is nothing in the law as it currently stands in this jurisdiction which supports this limb of the analysis.

Summary of the jurisdictions

It is clear from the current state of patent law in all three jurisdictions, that at best, only half of the framework is being considered. In the EU, under the “any hardware” approach, the framework is almost completely ignored.

Some commentators are particularly dismissive of *Bilski*’s machine-or-transformation test.²⁷ The Supreme Court was itself at least wary of the test, in that it did not endorse it as the sole test of patentability. Yet, as has been argued in this thesis, some form of physicality requirement has informed the scope of patent law since its inception. Physicality has also been of particular value in the software patent debate, as it provides a rational framework upon which to draw distinctions between “pure” software patents which would undermine free access to knowledge on the one hand (like the *Macrossan* invention),²⁸ whilst still allowing claims to physical devices and methods which include computable processes (like the *Aerotel* claims)²⁹ to be patentable.

Therefore it is to be expected that the physicality dimension has been the primary one for patent law, to this point. However, the other two dimensions,³⁰ have been shown to be particularly useful in that they move beyond an exploration of the nature of the thing claimed, to look at the nature of the process involved in creating it. This change in focus was demonstrated to be key to understanding the reason why mathematics was not patentable, since the competing claims as to the nature of mathematics amount to something of a dead end. Similarly, these two aspects when considered in the context of software defied what might be considered a common

²⁴ Advisory Council on Intellectual Property, *Patentable Subject Matter* (Final Report, December 2010) at 13.

²⁵ Advisory Council on Intellectual Property, *Patentable Subject Matter* (Final Report, December 2010) at 13.

²⁶ *Grant v Commissioner of Patents* (2006) 154 FCR 62, discussed in Chapter 3 at 128.

²⁷ Lemley et al describe the test as “deeply flawed”: Mark A Lemley et al “Life After *Bilski*” (2011) 63 *Stanford Law Review* 1315 at 1316. See also Dennis Crouch and Robert P Merges, “Operating Efficiently Post-*Bilski* by Ordering Patent Doctrine Decision-Making” (2011) 25 *Berkeley Technology Law Journal* 1673 at 1690: “the subject matter eligibility test under *Bilski* is quite difficult for anyone to implement (because of the lack of guidance), and on a comparative basis, an examiner’s time is better spent applying the other patentability doctrines and at least temporarily ignoring subject matter eligibility questions.”

²⁸ *Aerotel Ltd v Telco Holdings Ltd and in the Matter of Macrossan’s Application* [2006] EWCA Civ 1371 at [58]-[74], discussed in Chapter 3 at 135.

²⁹ *Aerotel Ltd v Telco Holdings Ltd and in the Matter of Macrossan’s Application* [2006] EWCA Civ 1371 at [50]-[57], discussed in Chapter 3 at 4.

³⁰ Expressive or functional; aesthetic or rational.

understanding of the nature of software, and demonstrated why programming was more an artistic than an engineering activity. As such, the framework illustrates a shortcoming in current approaches to patentability. Confusing ontological issues might be replaced, or at least complemented, by a consideration of the creative process, to determine the appropriateness of the subject matter for the award of patent rights.

4 Broader implications of the argument for patent law

Subject matter is not a “failed gatekeeper”

The subject matter inquiry is a complex one. It involves a multitude of hard to reconcile cases, many and changing “investigative tool[s]”³¹ and a number of historical exclusions. Determinations of what is patentable and what is not depend on a comprehensive understanding of the nature of particular fields of technology as well as the operation of particular inventions. Unsurprisingly then, it may seem like a good solution to: take a broad view of eligible subject matter;³² reduce subject matter to a mere form requirement;³³ expressly disavow the subject matter inquiry in favour of the other technical requirements;³⁴ narrow the subject matter inquiry to a narrow technical one;³⁵ or avoid the issue by determining subject matter issues as a matter of last resort.³⁶ Whilst Lemley et al are right to draw attention to the difficulties attendant on drawing bright lines between patentable and non-patentable subject matter, “[t]wilight does not invalidate the distinction between night and day”.³⁷

The argument as to why software ought not be considered patentable need not be restated again here. However, the need for subject matter exclusions is also evident when alternative solutions as to the software patent problem are considered. In light of the arguments advanced in the last chapter, the suitability of these reforms as solutions to the software patent problem are considered below.

Alternative solutions are insufficient

Stricter disclosure requirements By enforcing the disclosure requirement much more strictly, it is said the scope of patent claims could be considerably narrowed. It is clear from the anal-

³¹ *Bilski v Kappos* 130 S Ct 3218 (2010) at 3227.

³² This is the effect of “useful result” approach, adopted in *State Street* in the US, and *Catuity* in Australia. As Lemley et al suggest, in the wake of the *State Street* decision, “patentable subject matter was effectively a dead letter”: Mark A Lemley et al, “Life After *Bilski*” (2011) 63 *Stanford Law Review* 1315 at 1318. A broad view is often indirectly adopted when appeal is made to the need for flexibility, or to adapt (or remove) traditional limitations in the wake of new “technologies”: see for example *Bilski v Kappos* 130 S Ct 3218 (2010) at 3227-3228 per Kennedy J, writing for a plurality. Whilst such a view is consistent with the broad view of technology as discussed in Chapter 6, at 5, both as a matter of intuition and as a matter of precedent, all human activity which advances the sum of human knowledge is not correctly described as “technology”.

³³ This is the approach adopted in the European Patent Office, wherein any claim to hardware is sufficient to pass muster under Article 52 of the European Patent Convention. See Chapter 3, Section 3 on page 115.

³⁴ See for example Michael Risch, “Everything is Patentable” (2008) 75 *Tennessee Law Review* 591.

³⁵ Mark A Lemley et al, “Life After *Bilski*” (2011) 63 *Stanford Law Review* 1315.

³⁶ Dennis Crouch and Robert P Merges, “Operating Efficiently Post-*Bilski* by Ordering Patent Doctrine Decision-Making” (2011) 25 *Berkeley Technology Law Journal* 1673.

³⁷ Murray Gleeson, “Judicial legitimacy” (2000) 20 *Australian Bar Review* 1 at 8.

Chapter 8: Implications

ysis in Chapter 7 that to truly describe the invention, the description of a computable process must be at a much lower level down the waterfall than at the design level (that is algorithms and data structures). Firstly, the physicality of the invention needs to be evident in order to satisfy the physical/intangible limb – it should be clear how this physicality limits the claims. But to properly describe a computable process, and in recognition of the need to cross the semantic gap, it is at least arguable that there should be disclosure of source code, or perhaps pseudocode.³⁸

On the positive side, such an approach might limit the effective life of the patent to a period consistent with the market life of software innovations. A more extensive disclosure of the invention could also result in better quality documentation of the prior art in the field, and increase the likelihood of patent literature becoming a useful source of technical information for programmers.

One problem with narrowing patent scope however, is that with computer programs, it is “quite possible to produce functionally indistinguishable program behaviours through the use of more than one method.”³⁹ Thus such a solution replaces an over-protection problem with one of under-protection. But the source code disclosure route must ultimately fail because it does not answer the criticism that programming involves a fine art, a creative process, and is not the sort of activity likely to be incentivised by the grant of patent rights.

Improved examination procedures Other solutions focus on the possibility of improving patent quality by improving the patent examination process. There have been moves in the US to improve funding to the USPTO⁴⁰ and address the high attrition rate of patent examiners.⁴¹ Given the ever-increasing presence of software in daily life and its accompanying growth in complexity, one should wonder whether patent office budgets can keep up. Certainly, despite some advances in the last 5 years, the backlog is still significant.⁴²

In 2005 the USPTO implemented a system of peer review of patent applications as a way of providing greater scrutiny by creating “a peer review system for patents that exploits network

³⁸ Pseudocode is an informal way of describing the algorithm behind a computer program independent of the syntactic constraints of a programming language. See for example Wikipedia, “Pseudocode” <<http://en.wikipedia.org/wiki/Pseudocode>> (8 August 2011).

³⁹ Pamela Samuelson et al, “A Manifesto Concerning the Legal Protection of Computer Programs,” (1994) 94 *Columbia Law Review* 2308 at 2345.

⁴⁰ Sarah L Stirland, “Bush makes new push for patent office to keep fees” *GovExec.com*, 6 February 2006, <http://www.govexec.com/story_page.cfm?articleid=33317> (15 August 2011). On the efforts to manage the backlog of applications, see Courtenay Brinkerhoff, “USPTO Backlog Update: One Step Forward, One Step Back” on *PharmaPatents* <<http://www.pharmapatentsblog.com/patent-office-practice/uspto-backlog-update-one-step-forward-one-step-back/>> (16 August 2011); United States Patent and Trademark Office “Backlog Stimulus Reduction Plan” <http://www.uspto.gov/patents/init_events/PatentStimulusPlan.jsp> (17 August 2011). The current size of the backlog stands at around 700,000 applications: United States Patent and Trademark Office, “July 2011 Patents Data, at a Glance” <<http://www.uspto.gov/dashboards/patents/main.dashxml>> (15 August 2011).

⁴¹ Olsen F, “Patent examiners battle stress,” *Information Today*, 25 July 2005, <http://fcw.com/articles/2005/07/25/patent-examiners-battle-stress.aspx?sc_lang=en> (15 August 2011).

⁴² As at July 2011, the backlog stands at 689,226 pending applications, with the average time from filing to a “first action” by the USPTO being 27.8 months, and a total average lag between filing and resolution of 33.5 months: USPTO, “July 2011 Patents Data, at a Glance” <<http://www.uspto.gov/dashboards/patents/main.dashxml>> (27 August 2011). Looking at the average filings since April 2011, it can be calculated that during that period, over 7,000 applications are filed each week.

Broader implications of the argument for patent law

technology to enable innovation experts to inform the patent examination procedure.”⁴³ Whilst such a reform is a positive step towards filling the gap in the documented prior art base, it is not without problems. One commentator has voiced concerns about the ability of such a system to deal with the huge volume of patent applications.⁴⁴ Based as it is on social software such as Wikipedia,⁴⁵ the system must also deal with similar concerns as to the potential for fraud and bias within such a system, particularly since patent applicants have more at stake than a favourable review in an online encyclopaedia.⁴⁶ It is also relevant to note that such a system is national in nature and thus depends on having a sufficient number of suitably qualified expert volunteers with a country’s borders, or else relying on people from other jurisdictions who may not be aware of local innovations.

All this aside, perhaps the biggest limitation of Peer to Patent is that it will only apply to the examination of up-and-coming patents. This means that the current crop of bad patents will dominate the software development landscape for the next 20 years.

Ultimately, it is submitted that improved examination procedures, whilst always to be welcomed, amount to little more in this context than re-arranging the deck chairs on the Titanic. As with the disclosure reforms, such changes don’t answer the criticism that this is not “inventive” activity, in the sense of being within the useful arts.

Compulsory licensing Compulsory licensing is one means of addressing the problems which the power to exclude causes in an industry characterised by sequential innovation and a high component-to-product ratio. Theoretically, the availability of a compulsory licence limits the ability of a patent holder to hijack a competitor’s business by refusing to licence a patented invention, by allowing the competitor to obtain a licence from the Crown. However, the need to pay a ‘reasonable’ licence fee, and a requirement of a court order mean the transaction costs in using such a scheme are impracticably high. Further, compulsory licences do little to improve the quality or scope of awarded software patents.

Exemptions from infringement Other solutions borrow from the copyright approach, in order to recognise a distinction between unfair imitation and legitimate reverse engineering.

⁴³ “About Community Patent,” The Community Patent Project, <<http://dotank.nyls.edu/communitypatent/about.html>> (4 June 2006). For more detail, see Beth Noveck, “‘Peer to Patent’: Collective Intelligence and Intellectual Property Reform” *NYLS Legal Studies Research Paper* No. 05/06-18, 25 April 2006, <<http://ssrn.com/abstract=898840>> (23 August 2006).

⁴⁴ Jason Schultz, attorney for the Electronic Frontier Foundation. Quoted in Daniel Terdiman, “Web Could Unclog Patent Backlog,” *Wired News*, 14 July 2005 <<http://www.wired.com/news/technology/0,68186-0.html>> (23 August 2006).

⁴⁵ Jason Schultz, attorney for the Electronic Frontier Foundation. Quoted in Terdiman D, “Web Could Unclog Patent Backlog,” *Wired News*, 14 July 2005, <<http://www.wired.com/news/technology/0,68186-0.html>> (23 August 2006).

⁴⁶ See R Demsyn, “Wal-marts Wikipedia War,” on *Whitedust Security*, 28 April 2006, <http://www.whitedustnet/article/55/Wal-marts_Wikipedia_War/> (23 April 2006); “Wikipedia Criticised By Its Co-Founder,” on *slashdot.org*, 3 January 2005, <<http://slashdot.org/articles/05/01/03/144207.shtml>>; David Mehegan, “Bias, sabotage haunt Wikipedia’s free world,” *The Boston Globe*, <http://www.boston.com/news/nation/articles/2006/02/12/bias_sabotage_haunt_wikipedias_free_world/>. Despite such criticisms, this author continues to cite Wikipedia, even in this thesis, when the author feels the referenced material can be confirmed by external sources, and explains the material well.

Chapter 8: Implications

O'Rourke advocates creating a 'fair use' defence, along the lines of that available in US copy-right law.⁴⁷ According to this reform, courts would have the power to balance innovation incentives against social benefits in the context of a particular case to determine whether an otherwise infringing activity should be allowed. Such a reform could perhaps bring the patent system more into line with cumulative innovation, and to acknowledge the importance of openness to software development. However, there is a fair amount of uncertainty inherent in the operation of this scheme in that the party claiming fair use must be prepared to brave a patent infringement suit in order to establish their entitlement to continue in their activity.

Another possibility is to explicitly limit the interpretation of patent claims so as to exclude independently implemented functional equivalents.⁴⁸ This may be another path to the same objective as the stricter disclosure reform outlined above. Thus it shares the likelihood that it will result in the under-protection of software inventions.

Private initiatives A number of private initiatives also seek to address the patent problem. IBM, in 2005, started the ball rolling by pledging not to enforce 500 of the patents in its portfolio against Free and Open Source Software (FOSS) developers, and promised to enforce their patents against any organisation pursuing infringement action against an OSS project. Similar pledges have since been made by Sun, Nokia, Novell, Red Hat and Computer Associates.⁴⁹ Putting aside questions of whether these pledges are just publicity stunts,⁵⁰ such moves are to be applauded, in that they provide open source developers with at least some level of protection against infringement suits. The problem with such initiatives is that they cannot prevent so-called 'patent trolls' from enforcing their patents against FOSS developers. The typical defensive patent portfolio strategy, which such schemes extend to the FOSS community, is ineffective against trolls, since they do not generally engage in software development and are hence safe from cross-claims for infringement.

Another initiative seeks to raise standards in future software patents by addressing the prior art gap. The Open Source as Prior Art project⁵¹ hopes to improve patent quality by "improving accessibility by patent examiners and others to electronically published source code and its related documentation as a source of prior art."⁵² This is a commendable effort to address one of the root causes of poor patent quality in software, yet it is far from a complete solution, and it can do nothing to resolve the problems with dubious patents which have already been awarded.

A final strategy directs the global collaboration model of FOSS development at defeating bad patents. The Public Patent Foundation⁵³ is a not-for-profit organisation which volunteers to

⁴⁷ Maureen A O'Rourke, 'Toward a Doctrine of Fair Use in Patent Law' (2000) 100 *Columbia Law Review* 1177.

⁴⁸ Julie E Cohen and Mark A Lemley, 'Patent Scope and Innovation in the Software Industry' (2001) 89 *California Law Review* 1 <<http://www.law.georgetown.edu/faculty/jec/softwarepatentscope.pdf>> (2 November 2004).

⁴⁹ Robin Cover (ed), "Open Source Development Labs (OSDL) Announces Patent Commons Project" *Cover Pages*, 10 August 2005, <<http://xml.coverpages.org/ni2005-08-10-a.html>> (13 September 2011).

⁵⁰ The motives of IBM in particular should be questioned, given the small size of their pledge compared to the size of their portfolio, and the prominence of their role in pushing the pro-IP big business agenda. On the latter point, see Drahos, (ref) at 169-173.

⁵¹ <<http://www.linuxfoundation.org/programs/legal/osapa>> (27 August 2011).

⁵² "Overview," <<http://osapa.org/information.html>> (4 June 2006).

⁵³ See <<http://pubpat.org/>> (18 January 2005).

Broader implications of the argument for patent law

search for prior art which will invalidate bad patents. The Foundation has had some successes in challenging patents on the JPEG image format,⁵⁴ the drug Lipitor⁵⁵ and the FAT filesystem.⁵⁶ However, this is a mere drop in the vast ocean of software patents, and amounts to little more than “swatting mosquitoes to cure malaria”.⁵⁷

Summary With the exception of the examination improvements, the reforms discussed so far also face a deeper problem. They are technology-specific solutions aimed at tweaking the patent system to meet the needs of the software industry. On a practical level, the consequence of this is that these reforms are likely to face strong opposition from the powerful lobby groups of other industries where the patent system appears to work well, most notably the pharmaceutical industry.

Giving effect to the purpose of patent law

Patentable subject matter is also a necessary part of patent law because of the way it gives effect to the purpose of patent law. This purpose is to “benefit society through optimising innovation and public access to new technologies.”⁵⁸ As the strongest form of monopoly rights, patents can be either significant incentives to innovate, or significant barriers to competition in an industry. As an exception to a general distaste for monopolies, patents should only be available where such a purpose can be served.

Patents are sometimes justified on purely economic grounds. Patent theory asserts that “where the norm of free competition would result in free riding by competitors and less reason to invest in new technologies, innovation is encouraged by providing innovators with the exclusive rights to their inventions.”⁵⁹ Innovation is considered to be a positive on the basis that the public benefit from the disclosure of new and useful technologies, which fall into the public domain at the expiry of the patent grant period – the social contract theory of patent law.⁶⁰ These benefits must be contrasted with the award of monopoly rights, which amount to a tax on society

⁵⁴ See <<http://pubpat.org/Chen672Rejected.htm>> (1 June 2006).

⁵⁵ See <<http://pubpat.org/LipitorPatentNarrowed.htm>> (1 June 2006).

⁵⁶ See <http://pubpat.org/Microsoft_517_Rejected.htm> (1 June 2006).

⁵⁷ Richard Stallman, ‘The Danger of Software Patents’ 2004 Cyberspace Law and Policy Seminar (audio recording), Sydney, 14 October 2004 <<http://images.indymedia.org/imc/sydney/stallman%20lo.ogg>> (11 November 2004).

⁵⁸ Advisory Council on Intellectual Property, *Patentable Subject Matter* (Issues Paper, July 2008) at 1.

⁵⁹ Advisory Council on Intellectual Property, *Patentable Subject Matter* (Issues Paper, July 2008) at 1.

⁶⁰ This line of thinking emerged *Liardet v Johnson* (1778) 1 Carp Pat Cas 35 (NP), discussed in Chapter 2 at 45. See also *Turner v Winter* (1787) 19 Eng Rep 1276 where the Court noted that “[t]he consideration, which the patentee gives for his monopoly, is the benefit which the public are to derive from his invention after his patent is expired”. However, economic rationales for patent law began to dominate as the eighteenth century progressed. See Edward C Walterscheid “The Early Evolution of the United States Patent Law: Antecedents (Part 4)” (1996) 78 *Journal of the Patent and Trademark Office Society* 77 at 104-106. Similarly see the second reading speech for the *Patents Act 1990* Australia: “[t]he essence of the patent system is to encourage entrepreneurs to develop and commercialise new technology.”: Commonwealth of Australia, “Patents Bill 1990: Second Reading” Senate, 29 May, 1990 <http://parlinfoweb.aph.gov.au/PIWeb/view_document.aspx?id=562046&table=HANSARDS> (2 November 2004). A different theory underlies the US system, namely patents as a reward of inventors: Edward C Walterscheid “Patents and Manufacturing in the Early Republic” (1998) 80 *Journal of the Patent and Trademark Office Society* 855 at 856.

Chapter 8: Implications

in two different ways: first, by the high price of goods, which, in the case of a free trade, they could buy much cheaper; and, secondly, by their total exclusion from a branch of business which it might be both convenient and profitable for many of them to carry on.⁶¹

A narrow view of this social contract theory frames the patentable subject matter inquiry in purely economic terms. Yet there is nothing in social contract theory which requires that the benefit received by the public is be limited to economic, or even strictly utilitarian matters. It is submitted that in fact it is impossible to exclude these considerations from patent law, and attempting to do so only converts relevant policy considerations into inherent assumptions.⁶² The role of policy in patent law is clearest in Europe, where the various categories of exclusion were considered by the framers to contain their own important policy issues.⁶³ This point was noted by Laddie J in *Fujitsu*,⁶⁴ although his Honour's approach was ultimately rejected on appeal.⁶⁵ Pila advocates for the adoption of such an acknowledgment,⁶⁶ noting that it is "consistent with the increasing support for tribunals' reliance on external rules, including fundamental rights, in an effort to stem "the continuing expansion of intellectual property rights outside their traditional bounds" and thereby "correct the slide towards protection [and] reestablish the proper balance of interest."⁶⁷ Similarly, Burk and Lemley note the influence of technology specific policy considerations in the development of patent law in the US.⁶⁸

Whether considered on a broad or narrow view, one purpose of patent law is to promote innovation. Whether patent grants are appropriate in a particular field depends on whether that purpose can be achieved. The subject matter test stands apart from the technically-focused

⁶¹ Adam Smith, *An Inquiry into the Nature and Causes of the Wealth of Nations* (5th ed, Methuen, 1904) at 159-160.

⁶² "[R]ather than asking whether and how patent rights actually encourage inventive activity and dissemination of innovations, courts have simply presumed that they do so in the absence of, and sometimes against, actual evidence.": Richard Gold, "The Reach of Patent Law and Institutional Competence," (2003) 1 *University of Ottawa Law and Technology Journal* 263 at 277. This is what Gold calls "stealth libertarianism". On this approach, difficult questions of policy are reduced to mere technical questions of statutory interpretation. The *Bilski* majority is a case in point.

⁶³ Justine Pila, "Dispute over the Meaning of 'Invention' in Article 52(2) EPC - The Patentability of computer-implemented Inventions in Europe," (2005) 36 *IIC: International Review of Intellectual Property and Competition Law* 173 at 185.

⁶⁴ "The types of subject-matter referred to in section 1(2) are excluded from patentability as a matter of policy. This is so whether the matter is technical or not.": *Fujitsu Ltd's Application* [1997] RPC 608 at 614. See the discussion of the case in Chapter 3 3 on page 111.

⁶⁵ *Fujitsu Ltd's Application* [1997] RPC 608 (Court of Appeal) per Aldous LJ.

⁶⁶ Justine Pila, "Dispute over the Meaning of 'Invention' in Article 52(2) EPC - The Patentability of computer-implemented Inventions in Europe," (2005) 36 *IIC: International Review of Intellectual Property and Competition Law* 173 at 183-187.

⁶⁷ Justine Pila, "Dispute over the Meaning of 'Invention' in Article 52(2) EPC - The Patentability of computer-implemented Inventions in Europe," (2005) 36 *IIC: International Review of Intellectual Property and Competition Law* 173 at 185, citing Christophe Geiger, "'Constitutionalising' Intellectual Property Law? The Influence of Fundamental Rights on Intellectual Property in the European Union" (2006) 37 *International Review of Intellectual Property and Competition Law* 371.

⁶⁸ For example, Burk and Lemley note "an increasing divergence between the rules themselves and the application of the rules to different industries": Dan L Burk and Mark A Lemley, "Is Patent Law Technology Specific?" (2002) 17 *Berkeley Tech. Law Journal* 1155 at 1158, and that and that the acknowledgement of that fact requires "courts to build industry-sensitive policy analysis into their decisions": Dan L Burk and Mark A Lemley, "Policy Levers in Patent Law" (2003) 89 (7) *Virginia Law Review* 1575 at 1630. Cf R Polk Wagner, "Of Patents and Path Dependency: A Comment on Burk and Lemley" (2003) 18 *Berkeley Technology Law Journal* 1341, wherein the author acknowledges the technology specificity, but denies that it should be considered at anything other than a per-invention, or "micro" level.

tests of inventive step, novelty and utility, in that it alone is directed to the appropriateness of assuming the patent grant will enhance innovation in the field.

In this context, the three-dimensional framework has the advantage that it encourages a direct examination of the nature of the creative process that gives rise to the claims, to assess whether that process is of the kind which is likely to be enhanced by the award of patent rights. As such, it is an analysis which increases the likelihood that the purposes of patent law will be achieved.

The framework has a further important aspect. Despite the “excitingly unpredictable”⁶⁹ nature of patent law, the need for flexibility, and however much “times change”⁷⁰ it is suggested that this conservative approach, centred around a narrow understanding of the scope of technology, is a proper way to consider the subject matter issue.⁷¹ Whilst it is important that patent law be flexible enough to meet new circumstances, it should not be assumed that patent law is “like a nose of wax which may be turned and twisted in any direction”,⁷² without consequence. As the history of patentable subject matter set out in Chapter 2 discloses, and the analysis of the physicality of mathematics and software in Chapters 6 and 7 respectively show, the physicality requirement is both a link to the traditional foundation of patent law, and a necessary limiting device in contemporary understandings of patentability. Similarly, the distinction between the useful and fine arts is a long standing one, which ought to be maintained. If the non-physical, expressive and aesthetic arts of the Information Age are to be protected, it is not at all clear, nor should it even be presumed, that patent law provides the best mechanism to do so. It may well be that the new paradigm deserves a new regime to protect it which is contoured to the new balance between freedom and control which such arts require to encourage advancement.⁷³ Warping an industrial age mechanism designed for industrial era invention is not the solution.

In light of this globally conservative approach, it is also suggested that conservatism at the micro-level is appropriate. Working from a conservative standpoint here means that new areas should be presumed not to be patentable. Such an approach has a number of advantages. It puts the burden of proof of the need for patent protection on the well-organised, well-represented business organisations who are the primary customers of patent offices, instead of the unorganised and largely unrepresented public. It would recognise the cumulative nature of innovation in the knowledge economy, and signal a return to the notion that the “the noblest of human productions—knowledge, truths ascertained, conceptions, and ideas—become, after voluntary communication to others, free as the air to common use.”⁷⁴ It would encourage public consultation and empirical evidence gathering about the impact of patenting in new areas on the public benefits and detriments *before* such patents began to be awarded. As Gold notes,

⁶⁹ *National Research Development Corporation v Commissioner of Patents* (1959) 102 CLR 252 at 271.

⁷⁰ *Bilski v Kappos* 130 S. Ct. 3128 at 3227 per Kennedy J (writing for a plurality).

⁷¹ Cf *Bilski v Kappos* 130 S. Ct. 3218 (2010) at 3228 per Kennedy J, noting that “new technologies may call for new enquiries”, and proposing to modify the patent regime to accommodate the new Information Age.

⁷² *White v Dunbar* 119 US 47 at 51.

⁷³ The merits of a *sui generis* regime for software are considered below.

⁷⁴ *International News Serv. v. Associated Press*, 248 U.S. 215, 250 (1918) (Brandeis, J., dissenting). On a contemporary defence of this position, see Yochai Benkler, “Free as the Air to Common Use: First Amendment Constraints on Enclosure of the Public Domain,” (1999) 74 *New York University Law Review* 354 <<http://www.yale.edu/lawweb/jbalkin/telecom/benklerfreeastheairtocommonuse.pdf>> (15 December 2006); James Boyle, “The Second Enclosure Movement and the Construction of the Public Domain” (2003) 66 *Law and Contemporary Problems* 33.

Chapter 8: Implications

for courts to do otherwise is to assume competence over difficult questions of policy that they are ill-equipped to address.⁷⁵ Should a conservative interpretation of the scope of patent law create a gap which would-be patent applicants suggest needs to be filled, it is appropriate that either Parliament, or an administrative authority such as that considered above, undertake the relevant inquiry into the appropriateness of allowing patents in a new area.

Characterisation is key

The main weakness of the three dimensional framework is one that it shares with any subject matter test – potential mischaracterisation. The effectiveness of any analysis of an invention will depend on an accurate assessment of what the applicant has actually invented. Proper characterisation, it is submitted, depends on looking behind the form of the claims to their substance,⁷⁶ to answer the question from *Grams*, “what did [the] applicants invent?”⁷⁷

This seems to be handled both best and worst under European Patent Convention. The worst approach, being one which exalts form over substance, is that of the European Patent Office, who in allowing claims involving “any hardware” seem to take the word of the applicant as to what they have invented. The best approach is in the UK, where the list of excluded matter, although problematic, forces the recognition that “[i]t cannot be permissible to patent an [excluded item] under the guise of an article which contains that item”⁷⁸ To the extent that it is necessary to set out a proper approach to subject matter, a modification of the *Aerotel/Macrossan case*⁷⁹ is a good starting point:

1. Properly construe the claims
2. Identify the actual contribution
3. Ask whether it falls solely within the excluded subject matter
4. Check whether the actual or alleged contribution is actually technical in nature.⁸⁰

It is in the fourth step that the three dimensional framework comes into play, the question of whether an alleged contribution is technical in nature being the same as asking whether it falls within the fine arts, or is a technology. It will only be in the EU that the third step is necessary, to take account of the express exceptions in the EPC.

⁷⁵ “[G]iven the multiple and multifarious competing interests at stake in determinations of patent eligibility, the judiciary lacks both the capacity and the competence to resolve such issues”: Richard Gold, “The Reach of Patent Law and Institutional Competence,” (2003) 1 *University of Ottawa Law and Technology Journal* 263 at 283.

⁷⁶ *Fujitsu Ltd’s Application* [1996] RPC 511 at 530-531 per Laddie J.

⁷⁷ *In re Grams* 888 F.2d 839 (1989) at 839. See also the various formulations summarised in Chapter 4, n on page 166. See also *CyberSource Corporation v Retail Decisions Inc* Appeal No 2009-1358 (Fed Cir, 2011) at 17: “Regardless of what statutory category (“process, machine, manufacture, or composition of matter,” 35 U.S.C. § 101) a claim’s language is crafted to literally invoke, we look to the underlying invention for patent-eligibility purposes.”

⁷⁸ *Merrill Lynch’s Application* [1989] RPC 561 at 569.

⁷⁹ *Aerotel v Telco Holdings & Others; In the matter of Macrossan* [2006] EWCA Civ 1371 at [40].

⁸⁰ *Aerotel v Telco Holdings & Others; In the matter of Macrossan* [2006] EWCA Civ 1371 at [40].

Generalisations *and* case-by-case analysis are required

Generally, speaking, there will be times where the desirability of awarding patents can be assessed for a whole field, but there will inevitably be exceptions in the form of boundary issues. The case in point in this respect is software. Although it may be clear what the process of developing software is, the growing ubiquity of computing has seen software components become commonplace in all manner of devices.⁸¹ Thus it may be possible for a particular innovation to be characterised in multiple ways, meaning that it falls “within” a number of subject matters. The European experience seems to confirm that even the most explicit blanket bans on patents in particular fields are not guaranteed to work, because of the ease with which alternative characterisations, creative drafting and generous interpretations of the statutory language work around them.

This is further exacerbated by the fact that the development of new fields of technology often arises in the context of existing areas of study.⁸² In this situation, it may not be practical to assess the patentability of an emerging field of science until it becomes ‘recognised’, by which time the award of broad patents over basic research in the field may have already done significant damage.⁸³ So to some extent, it will almost always be necessary for courts to determine patentability on a case-by-case basis. Even within the analysis of software in Chapter 7, it was conceded that there might be occasions where claims which might be broadly categorised as claims to software, could be patentable.

However, generalisations about particular types of inventions are instructive in two regards. Firstly, they allow a degree of certainty for innovators in that they can make some sort of assessment of the likelihood of patent protection being available before entering into the expensive and time consuming process of applying for a patent. A degree of certainty also benefits those tasked with examining the patent for validity. Secondly, they reflect the nature of judicial development of the law – courts determining the patentability of a claimed invention will always look for guidance from past cases, and will be influenced by the patentability of similar, or even analogical inventions. Since Australia is not a heavily litigated jurisdiction, this can lead to long periods in which the current state of the law is out of step with the technology it is supposed to regulate. For example, although the patentability of software first came to the attention of the US Supreme Court in 1972 in *Gottschalk v Benson*,⁸⁴ it was 1991 before the patentability of software was judicially determined in Australia in *IBM v Commissioner of Patents*.⁸⁵ In the meantime, it was left to the Australian Patent Office to provide guidance to patentees by up-

⁸¹ For example, is an innovative method of curing rubber, such as that in *Diamond v Diehr* which uses software to determine the heat of the mould an innovation in the field of software development, or in materials engineering?

⁸² The emergence of computer science from the field of mathematics is just one example.

⁸³ See for example, in relation to biomedical research, Michael A Heller and Rebecca S Eisenberg, “Can Patents Deter Innovation? The Anticommons in Biomedical Research” (1998) 280 *Science* 698 at 700-701. Cf Dianne Nicol and Jane Nielsen, “Patents and Medical Biotechnology: An Empirical Analysis of Issues Facing the Australian Industry” (Centre for Law and Genetics Occasional Paper No 6, 2003) at 89, noting empirical evidence that most respondents in their survey of the Australian biotechnology industry “were fairly optimistic about their ability to continue research despite the presence of broad patents, and felt that the problem of broad patents was dissipating as patent offices tightened up their examination procedures.”

⁸⁴ 409 US 63 (1973).

⁸⁵ (1991) 33 FCR 218.

Chapter 8: Implications

dating its Manual of Practice and Procedures to reflect the development of the law in other jurisdictions.⁸⁶

It may well be that the appropriate middle ground between case-by-case analysis and per-field determinations of technology lies in some form of administrative solution.⁸⁷ In this respect, the development of authoritative guidelines, rather than a pure legislative solution, is the appropriate way of moving forward. In addition to the appointment of bodies to undertake *ad hoc* reviews of an area,⁸⁸ expert advisory committees could (and should) be available to assist with the examination of difficult patents, the development of guidelines, and to conduct comprehensive reviews of areas of particular concern on an ongoing basis. Such bodies might properly consider the full range of economic, social and ethical considerations relevant to the subject matter issue to be resolved, engaging in public consultation as a part of the process. The decisions of this body may be subject to ordinary judicial review, and may also be given some statutory weight in judicial determinations of subject matter issues.

Human rights jurisprudence

There is a growing awareness of the tension between intellectual property rights and human rights such as rights to adequate health care, to education, to share in the benefits of scientific progress, and to participation in cultural life.⁸⁹ Such considerations are viewed by some as external to the concerns of the patent regime, and somehow irrelevant to the development of patentable subject matter principles.⁹⁰ However, it is submitted that many of these rights are fundamental to a proper understanding of the traditional limits on patentable subject matter. In Chapter 5, it was shown how the notions of freedom of thought and freedom of expression,

⁸⁶ As of August 2011, it seems that the Australian Patent Office is “updating” the Australian law in light of the developments in *Bilski*, in *Invention Pathways Pty Ltd* [2010] APO 10 (21 July 2010); and the recent Federal Circuit case of *CyberSource Corporation v Retail Decisions Inc* Appeal No 2009-1358 (Fed Cir, 2011) in *Network Solutions, LLC* [2011] APO 65 (19 August 2011). For a critical summary of these developments, see Mark Summerfield, “Australian Patent Office Shoots Down Another ‘Business Method’” on *Patentology* <<http://blog.patentology.com.au/2011/08/australian-patent-office-shoots-down.html>> (28 August 2011).

⁸⁷ “Once Congress and the courts have been eliminated, the only plausible remaining candidate for the job ... is an administrative agency”: William W Fisher III, *Promises to Keep: Technology, Law and the Future of Entertainment* (Stanford University Press, 2004) at 195.

⁸⁸ Despite the US Presidential Commission which recommended against software patenting in 1966 being made up of “distinguished scientists, academics, and representatives of leading computer and high technology firms (as well as the Commissioner of Patents)”: Pamela Samuelson, “Benson Revisited: The Case Against Patent Protection for Algorithms and Other Computer Program-Related Inventions,” (1990) 39 *Emory Law Journal* 1025 at 1038, the recommendations were never acted upon.

⁸⁹ See for example Laurence R Helfer, “Human Rights and Intellectual Property: Conflict or Coexistence?” (2003) 5 *Minnesota Journal of Law, Science and Technology* 47; Richard Gold, “The Reach of Patent Law and Institutional Competence,” (2003) 1 *University of Ottawa Law and Technology Journal* 263 at 283-284, noting the need to consider the social, ethical and economic implications of technology in order to guarantee its just use; James Boyle, “Enclosing the Genome: What the Squabbles over Genetic Patents Could Teach Us” in F Scott Kieff, *Perspectives on the Human Genome Project* (Academic Press, 2003) 97, recognising that the patenting of genes involves a range of considerations beyond a narrow utilitarian inquiry. The impact of patent grants on public health is recognised by the *Doha Declaration on TRIPS and Public Health*, WTO Doc WT/MIN (01)/DEC/2 (2001) which allows Members to flexibly interpret the patent obligations in TRIPS to allow for access to medicines.

⁹⁰ See for example *Grant v Commissioner of Patents* (2006) 154 FCR 62 at 72: “It is not relevant, in our view, that some may think that a method or product will not advance the public interest. Once a product or process has been patented, its use is subject to the laws of the land, such as (to take but a few examples) those concerned with environmental protection, pharmaceutical product approval and occupational health and safety.” See James Boyle, “Enclosing the Genome: What the Squabbles over Genetic Patents Could Teach Us” in F Scott Kieff, *Perspectives on the Human Genome Project* (Academic Press, 2003) 97 at 106-109.

fundamental rights so familiar to the human rights lawyer, inform an understanding of why mathematical advances are in the fine rather than the useful arts.

One possible approach in this thesis would have been to frame an argument against patentability based in human rights.⁹¹ That course was not taken. Nonetheless, it is clear that the framework bears more than a passing relationship to these freedoms. That this is so is obvious in the distinction drawn between expression and purpose. Where the fine arts are involved, the importance to be placed on freedom of expression is greater, as the expression itself is a source of value. Where an activity is purposive, expression must submit to function, and therefore the need for freedom of expression is lessened. In other words, the need for freedom in a particular discipline may aid in determining whether a particular art is better categorised as a fine art or a useful art. A similar point can be made in relation to the intangible/physical distinction. Where abstract concepts are involved, and the physical realisation of such concepts is only of secondary significance, freedom of thought is of fundamental importance. As a corollary, if an activity is directed by, or heavily reliant upon human thought, then it is more likely that a fine art is involved.

The aesthetic/rational dimension is more difficult to neatly characterise. Aesthetic responses are described as emotional, intuitive and highly subjective. This of itself suggests the importance of the individual, and in particular the individual mind. This of course invokes the need for freedom of thought. But as discussed in Chapters 6 and 7, such reactions are both provoked by expression, and aesthetics also motivates the creation of expressive works. It may be that the aesthetic operates at the intersection of expression and thought. As such the importance of thought and expression together will tend to support a classification of subject matter as falling within the fine arts.

Understanding that the freedom operates as a consideration in the patentable subject matter dispute naturally opens the door to a consideration of the intersection of intellectual property and human rights jurisprudence. It is acknowledged that “the historical connections between intellectual property and human rights are thin at best.”⁹² As noted above, justifications of the patent regime are often couched in economic or utilitarian terms. As a result, “patent law has been almost entirely isolated from First Amendment considerations”.⁹³ Yet to deny the applicability of human rights in the patent context is wrong, because it ignores one of the fundamentals of property law – that awarding property rights does not just give the holder a power to control the thing which is the subject of the grant, it also gives the holder a power over other people.⁹⁴

In Chapter 6,⁹⁵ it was noted in passing that a human rights characterisation of patent law is considered by some to be paradoxical; as an irreconcilable struggle “when one human right is pitted against another, when intellectual property rights are used to restrict access to information that could – at no real cost to the developer – be deployed in ways that satisfy fundamental

⁹¹ At an earlier stage of this thesis, just such an argument was envisaged.

⁹² Peter Drahos “Intellectual Property and Human Rights” [1999](3) *Intellectual Property Quarterly* 349 at 357.

⁹³ Dan L Burk, “Patenting Speech” (2001) 79 Tex. L. Rev. 99, at 137.

⁹⁴ See for example Morris R Cohen, “Property and Sovereignty” (1927) 13 *Cornell Law Quarterly* 8 at 13, who characterises property as a sovereign power: “[W]e must not overlook the actual fact that dominion over things is also *imperium* over our fellow human beings.”

⁹⁵ Section 2 on page 235.

Chapter 8: Implications

needs.”⁹⁶ However, rather than characterising the intersection as a battle between the conflicting spheres of human rights and intellectual property, it is submitted that it is of more benefit to look within the patent paradigm to determine how human rights values capture and explain the “multiple and multifarious competing interests at stake in determinations of patent eligibility”.⁹⁷

The trend towards a narrow inquiry, including an unwillingness to consider the effect of patent grants on human rights, is not evident in copyright law. In fact, copyright law has been claimed to be “the engine of free expression”.⁹⁸ The history of copyright in the US jurisdiction reveals an understanding of the links between copyright and the pursuit of democracy.⁹⁹ In 1790, the Senate committee noted that “literature and science are essential to the preservation of a free Constitution”¹⁰⁰ namely because of the way it supports the diffusion of knowledge amongst the electorate.

In addition, freedom of thought arguably finds its place at the boundaries, through the idea/-expression dichotomy,¹⁰¹ and the related doctrines of merger,¹⁰² and *scènes à faire*.¹⁰³ Both of the latter doctrines are of particular relevance to determining the copyrightability of software in the US, being key components of the second stage of the *Altai* abstraction-filtration-comparison test.¹⁰⁴

⁹⁶ Rochelle C Dreyfuss, “Patents and Human Rights: Where is the Paradox” in William Grosheide (ed), *Intellectual Property and Human Rights: A Paradox*, (Edward Elgar, 2010) 72 at 72.

⁹⁷ Richard Gold, “The Reach of Patent Law and Institutional Competence,” (2003) 1 *University of Ottawa Law and Technology Journal* 263 at 283. See also Rochelle Dreyfuss, “Patents and Human Rights: Where is the Paradox” in William Grosheide (ed), *Intellectual Property and Human Rights: A Paradox*, (Edward Elgar, 2010) 72 at 73: “As a theoretical matter, there are clearly human rights concerns (rights to protect one’s dignity, to be compensated for one’s labor, and to enjoy one’s property without arbitrary governmental interference).”

⁹⁸ *Harper & Row, Publishers, Inc. v Nation Enters* 471 U.S. 539 (1985) at 558.

⁹⁹ As noted in Chapter 6, one of the justifications of freedom of expression is that it promotes democracy. See Chapter 6, at 247 and following.

¹⁰⁰ Bruce W. Bugbee, *Genesis of American Patent and Copyright Law* (1967) at 137, cited in Neil Weinstock Netanel, “Copyright and a Democratic Civil Society” (1996) 106 *Yale Law Journal* 283 at 289.

¹⁰¹ See *Baker v Selden* 101 U.S. 99 (1879) at 100-01: “[T]he truths of a science or the *methods of an art* are the common property of the whole world, and [the] author has the right to express the one, or explain and use the other, in his own way.” (emphasis added); *Hollinrake v Trusswell* [1894] Ch 420 at 427 per Lindley LJ: “Copyright ... does not extend to ideas, or schemes, or systems, or methods; it is confined to their expression; and if their expression is not copied the copyright is not infringed.” In *Autodesk v. Dyason* (No.2) (1993) 176 CLR 300, Mason J described the idea/expression divide as “the dominant principle in copyright law”. Dawson J in the same case noted that “when the expression of any idea is inseparable from its function, it forms part of the idea and is not entitled to the protection of copyright. See also *IceTV v Nine Network* (2009) 239 CLR 458 at [22]-[28] per French CJ, Crennan and Kiefel JJ.

¹⁰² The doctrine of merger holds that where only a limited number of ways of expressing an idea exist, the expression merges with the idea and becomes unprotectable. See *Baker v Selden* 101 U.S. 99 (1880) at 103: “Where the art [that the book] teaches cannot be used without employing the methods and diagrams used to illustrate the book, or such as are similar to them, such methods and diagrams are to be considered as necessary incidents to the art, and given therewith to the public.”

¹⁰³ The doctrine of *scènes à faire* operates to remove from copyright “those elements which are standard and inevitably arise in the treatment of a given topic”: Stanley Lai, *The Copyright Protection of Computer Software in the United Kingdom* (Hart Publishing 2000) 41 at [3.1].

¹⁰⁴ See *Computer Associates Intl, Inc v Altai Inc* 982 F.2d 693 (1992) at 709-710. The court noted that the doctrine of merger is particularly relevant in the context of software, since:

a programmer’s freedom of design choice is often circumscribed by extrinsic considerations such as (1) the mechanical specifications of the computer on which a particular program is intended to run; (2) compatibility requirements of other programs with which a program is designed to operate in conjunction; (3) computer manufacturers’ design standards; (4) demands of the industry being serviced; and (5) widely accepted programming practices within the computer industry; such “external factors” may make expression unprotectable.

Even in Australia, with its limited conception of freedom of speech,¹⁰⁵ the idea versus expression dichotomy has been directly linked with freedom of expression. In *Skybase Nominees*, Hill J in particular noted the

tension in policy between the monopoly rights which are conferred upon the owner of copyright in a literary, dramatic or artistic work on the one hand, and the freedom to express ideas or discuss facts on the other. While there will be an infringement of the copyright of an owner in a literary, dramatic, musical or artistic work where there is a reproduction of that work or a substantial part of it, the fact that another work deals with the same ideas or discusses matters of fact also raised in the work in respect of which copyright is said to subsist will not, of itself, constitute an infringement. *Were it otherwise, the copyright laws would be an impediment to free speech, rather than an encouragement of original expression.*¹⁰⁶

What is suggested in relation to the patent paradigm then is that there is plenty of room for human rights theory within the traditional bounds of the patent law paradigm. The fact that a framework such as the one developed, which is consistent with the theory of patent law, and which opens the door to such considerations can be developed, suggests that other evaluative tools might be similarly developed to address other ethical and social values which patent grants inevitably affect.

5 Beyond the patent paradigm

If software is to be pushed outside the patent regime, the question arises as to how software ought to be protected, if it is to be protected at all. Whilst an answer to that question could well provide enough material for another thesis, it is apposite to say something about how the software landscape might look in the absence of patent protection.

It is often assumed that inventors will have no way of stopping competitors from imitating their innovations. Although it is impossible to measure how many competitors would copy an unprotected innovation in the absence of patent protection, some indication of the likely extent of the problem can perhaps be gleaned by looking at the extent to which copying can be found in patent infringement cases. Despite proof of copying not being a requirement of infringement, it seems fair to conclude that those who would copy an invention *with* patent protection form a subset of those who would copy without patent protection. It might then be expected that a significant proportion of infringement actions would involve copying rather than independent invention. However, a 2008 study by Cotropia and Lemley¹⁰⁷ found allegations of copying

¹⁰⁵ Specifically, all that is recognised is an implied freedom of political communication, which acts not as a positive right but a limitation on legislative power. See *Lange v Australian Broadcasting Corporation* (1997) 189 CLR 520.

¹⁰⁶ *Skybase Nominees Pty Ltd v Fortuity Pty Ltd* (1996) 36 IPR 529 at 531 per Hill J (French J concurring) (emphasis added).

¹⁰⁷ Christopher Cotropia and Mark Lemley, "Copying in Patent Cases (draft)" <http://www.law.berkeley.edu/institutes/bclt/students/2008_ip-seminar/Lemley_Copying-in-Patent-Law1.pdf> (17 September 2008).

Chapter 8: Implications

(or facts that suggested that copying may be in issue) in 11% of cases,¹⁰⁸ and copying was actually found in only 1% of cases reviewed. This factor varied by industry, with the high water mark being the pharmaceutical industry, where allegations of copying were found in 65% of cases (no doubt due to the existence of generic producers). Computer-related inventions and software on the other hand, despite forming the largest subset of the cases considered, involved an extremely low rate of allegations of copying in only 2.6% and 3% of cases respectively.¹⁰⁹ So it seems that the free rider problem may be less of an issue than it is usually given credit for, at least outside the pharmaceutical industry, where the importance of patent protection is generally uncontested. These figures certainly indicate, if nothing else, that presumptions about the scale of the free rider problem cannot be generalised.

The size of the free riding problem will also depend upon the extent to which alternative protection regimes are available. Alternative protection is typically assumed to mean trade secret protection, although it may often be the case that copyright could also be available. This is certainly true in the software context, where those against software patenting often claim that copyright provides adequate protection.¹¹⁰ The availability of copyright as an alternative protection paradigm may explain the low rate of copying in patent infringement suits noted above. On this note, the possibility of establishing a software-specific protection regime is considered below.

Software itself falls to be protected by more than just the patent regime. There is a significant overlap between patent, copyright and trade secret protection. All three paradigms are used to varying extents by software developers, and software companies, to protect their creations. The nature of the protection afforded by those other regimes, and the shortcomings of these protections is considered below.

Trade secrets

Trade secrets are sometimes treated as the antithesis of patent law. Patent law is said to encourage disclosure, where as trade secret law encourages secrecy. By distributing only object code, it is possible for the informational aspects of software to be obscured from view. So-called proprietary software depends on a combination of secrecy, and copyright to prevent literal copying. Various obfuscation techniques can also be used to make reverse engineering of soft-

¹⁰⁸ Christopher Cotropia and Mark Lemley, "Copying in Patent Cases (draft)" <http://www.law.berkeley.edu/institutes/bclt/students/2008_ip-seminar/Lemley_Copying-in-Patent-Law1.pdf> (17 September 2008), at 24.

¹⁰⁹ Christopher Cotropia and Mark Lemley, "Copying in Patent Cases (draft)" <http://www.law.berkeley.edu/institutes/bclt/students/2008_ip-seminar/Lemley_Copying-in-Patent-Law1.pdf> (17 September 2008) at 27.

¹¹⁰ See for example the No Software Patents! website, which explains: "Software developers are perfectly protected without patents. Everyone who writes a computer program automatically owns the copyright in it. It's copyright law that made Microsoft, Oracle, SAP and the entire software industry so very big. It's the same legal concept that also protects books, music, movies, paintings, even architecture.": *NoSoftwarePatents.com*, "The Basics" <<http://www.nosoftwarepatents.com/en/m/basics/index.html>> (4 August 2011).

ware more difficult,¹¹¹ as can various legal mechanisms, such as non-disclosure agreements, and anti-reverse engineering clauses in software licences.¹¹²

The time it takes software to be reverse engineered by competitors provides a definite first mover advantage to innovators, who can in theory move on to the second iteration of a product. That lead time can itself act as an incentive for the first-mover to create further refinements, which mean that their competitors are always one step behind. Further, the very act of reverse engineering is seen to be “an essential part of innovation,”¹¹³ as it invites follow-on innovators into the field, which “could lead to significant advances in technology.”¹¹⁴

It is sometime claimed that the lead time has shortened to such a degree that trade secrets have ceased to be an effective way of ensuring an appropriate reward for innovative research and development.¹¹⁵ But the extent to which a redirection from patent to trade secret should be considered a negative consequence depends on the extent to which patent applications amount to a useful disclosure of the invention. In the software context, patent literature is rarely of use to programmers. that the main reason for this is that the abstract nature of patent claims means that they fall on the wrong side of the semantic gap. So it must be questioned whether the absence of patent protection for software-related inventions should be considered a loss at all.

Further the impact of reverse engineering must be considered. It may be that the lead time associated with reverse engineering is more attuned to the market cycle for products than the static 20 year time frame associated with patents. If it is, then the likelihood of disruption of next-generation inventions is not possible in the same way that submarine patents can disrupt an industry by providing protection for other products not within the contemplation of the original inventor, suggesting an overall economic gain may be achieved by such a redirection. Also, reverse engineering of a product may in fact lead to new information about the innovation being discovered which can lead to valuable improvements. Finally, the economic value of patent protection depends on the extent to which independent invention should be viewed as a valuable exercise in its own right – in some fields like software it may be no less than critical.

Regardless of the availability of patents for software, trade secret protection remains an important protective mechanism. It has advantages over both patent and copyright in that it is immediate, does not require registration, is potentially broader in scope, and unlimited in duration. It also has its own technologically-sensitive built-in regulation mechanism – reverse engineering. Its suitability is therefore largely determined by the time required for competitors to reverse engineer then re-implement a particular secret. In fact the trade secret paradigm

¹¹¹ For a selection of the sorts of methods used in software, “Code Obfuscation Literature Survey” on *Obfuscators.org*, 13 April 2008 <<http://www.obfuscators.org/2008/04/code-obfuscation-literature-survey.html>> (1 September 2011).

¹¹² Pamela Samuelson and Suzanne Scotchmer, “The Law and Economics of Reverse Engineering” (2002) 111 *Yale Law Journal* 1575 at 1660. See also Electronic Frontier Foundation, “Codders’ Rights Project Reverse Engineering FAQ” <<https://www.eff.org/issues/coders/reverse-engineering-faq>> (1 September 2011), noting that anti-reverse engineering clauses for software may be found in “End User License Agreements (EULA), terms of service notices (TOS), terms of use notices (TOU), a non-disclosure agreement (NDA)”.

¹¹³ See *Bonito Boats, Inc v Thunder Craft Boats, Inc* 489 U.S. 141 (1989) at 160.

¹¹⁴ See *Bonito Boats, Inc v Thunder Craft Boats, Inc* 489 U.S. 141 (1989) at 160.

¹¹⁵ Jerome Reichman, “Legal Hybrids Between the Patent and Copyright Paradigms” (1994) 94 *Columbia Law Review* 2432; Pamela Samuelson et al, “A Manifesto Concerning the Legal Protection of Computer Programs,” (1994) 94 *Columbia Law Review* 2308.

Chapter 8: Implications

is certainly a feature of the proprietary software sector, who rely on copyright protection to prevent literal copying, and trade secret protection, via object code.¹¹⁶

Copyright

As is clear from the analysis in the previous chapter, the fine arts are the traditional province of copyright law. As such, it makes sense to consider the extent to which copyright provides appropriate protection. It certainly protects against literal copying, and is entirely consistent with the freedoms required in an aesthetic, expressive field such as software development. On any reading, the nature of source code as akin to a literary work is hard to deny. Yet, to enable “full” copyright protection, it has been necessary to modify the definition of a literary work to include object code. This has resulted in somewhat of a distorting of the conceptual consistency of a literary work, as object code is not “intended to afford either information and instruction, or pleasure, in the form of literary instruction” to any person, and is unlike other literary works in that way. However, as noted in the previous chapter, the information which is embedded in such code is still available, although it may require decompilation in order to make it available, and at the very least contains a kernel of expression.

The major limit of copyright which prompted a push for the patenting of software still exists however. This limitation is that copyright paradigm only protects the literal text of software, and “the primary source of value in a program is its behaviour, not its text”.¹¹⁷ Although text and behaviour are directly linked, they are not identical. It is possible for any skilled programmer to “copy the behaviour of a program exactly, without appropriating any of its text.”¹¹⁸ In this situation, the ‘imitation’ will not infringe the copyright in either source or object code, but will behave in a manner indistinguishable to an end user from the original.

However, there is a significant proportion of global software development which depends only on copyright protection. That is the Free and Open Source software sector.¹¹⁹ Populated by a significant number of volunteers, but also a number of large commercial software development organisations, the FOSS model is different to the proprietary software camps in that it eschews secrecy in favour of the open disclosure of source code. Rather than seeking increased protection, the FOSS model actually increases the freedom available to developers by granting an open licence to reuse, distribute and modify, with the only real limitation being that such freedoms are to be passed downstream.¹²⁰ The significance of the FOSS sector was discussed in Chapter 4.¹²¹

¹¹⁶ In fact, it is not unusual to use code obfuscation as a technique to extend lead time. See for example, Wikipedia, “Obfuscated code” <http://en.wikipedia.org/wiki/Obfuscated_code> (11 September 2011). The camel code, and DeCSS code in Chapter 7 are examples of obfuscated code although the purpose was not to increase lead time, but for aesthetics.

¹¹⁷ Pamela Samuelson et al, “A Manifesto Concerning the Legal Protection of Computer Programs,” (1994) 94 *Columbia Law Review* 2308 at 2315.

¹¹⁸ Pamela Samuelson et al, “A Manifesto Concerning the Legal Protection of Computer Programs,” (1994) 94 *Columbia Law Review* 2308 at 2315.

¹¹⁹ See the discussion of FOSS software and its significant in Chapter 4 at 175.

¹²⁰ For an overview of the various alternatives, see Kenneth Wong and Phet Sayo, “Free/Open Source Software: A General Introduction,” *International Open Source Network*, 2004 <<http://www.iosn.net/foss/foss-general-primer/>> (18 September 2010).

¹²¹ See Chapter 4, on page 175.

Between the cracks?

To the extent that suitable protection is not offered neither patent, trade secret, or copyright, it is submitted that this is not a fault of patent law, but an inevitable consequence of the disparate nature of the regimes which fall under the penumbra of intellectual property. Where there is a consistent theoretical framework informing any intellectual property paradigm, it can maintain that consistency. And to an extent that consistency requires that inconsistent subject matter be excluded.

Copyright, according to the *Baker v Selden* exclusion excludes functional subject matter. So whilst software has caused some theoretical weakening of the consistent paradigm, by watering down the notion of a literary work to cover things not intended to afford literary instruction or enjoyment, functional protection is clearly excluded. Similarly, it has been shown that patent law ought to exclude the subject matter of the fine arts. This may mean that software fails to be afforded anything other than the “thin” protection of copyright law – against literal copying – but is denied any form of functional protection.

It may be suggested that such protection has marginal, if any value, and that copyright is falling through a crack between copyright and patent paradigms. To an extent, this may be true, in that neither paradigm is equipped to deal with subject matter that is both descriptive and functional. Given the independent history of each of the paradigms, that this might happen does not represent a failure of either paradigm. It merely suggests that it may be time to consider adding another paradigm to the intellectual property stable.

An alternative protection paradigm?¹²²

Over fifteen years ago, a group of leading computer science and intellectual property scholars turned their efforts to a “normative analysis of the kind of legal protection that would be socially desirable for software and how it might best be accomplished”.¹²³ The authors were critical of the way in which discussions of the software protection problem focused on adapting existing regimes, showing how such an approach leads to recurrent cycles of under- and over-protection.¹²⁴ Their conclusion was that a *sui generis* regime was the only way to correctly strike the balance between creating incentives to innovate and upholding the public interest.¹²⁵ Their proposed regime starts from the principle of “*market preservation*, that is, constructing just enough machinery to head off the ways in which marketplaces fail.”¹²⁶ The regime chosen to give effect to that principle is based around the following four precepts:

¹²² An earlier version of this section was published as Anton R Hughes, “Avoiding the Software Patent Problem: An Alternative Fix for TRIPS Junkies” (2007) 14(1) *eLaw: Murdoch University Electronic Journal of Law* 100.

¹²³ Pamela Samuelson et al, “A Manifesto Concerning the Legal Protection of Computer Programs,” (1994) 94 *Columbia Law Review* 2308.

¹²⁴ Pamela Samuelson et al, “A Manifesto Concerning the Legal Protection of Computer Programs,” (1994) 94 *Columbia Law Review* 2308 at 2356-2357.

¹²⁵ Pamela Samuelson et al, “A Manifesto Concerning the Legal Protection of Computer Programs,” (1994) 94 *Columbia Law Review* 2308 at 2356-2357.

¹²⁶ Randall Davis et al, “A New View of Intellectual Property and Software” (1996) 39 *Communications of the ACM* 21 at 21. Davis is one of the authors of the longer Manifesto article, and this article provides a useful summary of it.

Chapter 8: Implications

- Traditional copyright protection for literal code;
- Protection against behavior clones for a market-preserving period;
- Registration of innovation to promote disclosure and dissemination; and
- A menu of off-the-shelf liability principles and standard licenses.¹²⁷

Copyright protection is an important starting point in software, since it alone was sufficient for the early successes of the software industry.¹²⁸ Copyright protection, as the traditional protection regime for the fine arts, is an obvious starting point for an appropriate protection mechanism, given the result of the analysis conducted in Chapter 7. Because it protects innovators against free-riding imitators, whilst not interfering with independent creation, it promotes the freedom which lies at the core of software development, and upon which the analytical framework is based. Copyright also lies at the heart of the FOSS paradigm discussed above. Such a regime would allow FOSS projects to continue with little disruption.

But the absence of functional protection suggests copyright alone may be insufficient. In particular, the analysis of the nature of software in Chapter 7 noted that software contains functional elements, and that function may become a dominant consideration. The *sui generis* system directly addresses the behavioural protection issue because, like the patent regime, it provides some protection against functional imitation, or “behaviour clones”. The system works in a similar fashion to the trade secret paradigm by creating an artificial lead-time for the registrant of a grain-sized innovation, in exchange for disclosure of the details of the innovation. All those wishing to use the innovation during the lead-time must pay a royalty to the innovator, or else wait out the liability period.¹²⁹

These behavioural protections have been argued by Reichman to be equivalent to patent protection in that no one is worse off under this system than they would be if software patents were available.¹³⁰ Registrants are protected against functional imitation for a period consonant with the market life of their product. Follow-on innovators contribute to the research and development costs of upstream inventors, without the inconvenience of the large transaction costs of licence negotiation. Developers of independent products can stand on their own in the marketplace, without fear of ambush. In addition, the regime upholds the importance of aesthetics. It encourages disclosure, not through the patent claims and specification, written

¹²⁷ Randall Davis et al, “A New View of Intellectual Property and Software” (1996) 39 *Communications of the ACM* 21.

¹²⁸ See for example Richard Stallman, “For submission to the Patent & Trademark Office,” 1994, <<http://lpf.ai.mit.edu/Patents/rms-pto.html>> (13 September 2006); Bill Gates, “Challenges and Strategy” Microsoft Internal Memo, <<http://www.bralyn.net/etext/literature/bill.gates/challenges-strategy.txt>> (13 September 2006); D Brotz, (Principal Scientist, Adobe Systems), Public Hearing on Use of the Patent System to Protect Software Related Inventions, Transcript of Proceedings Wednesday, January 26, 1994 San Jose Convention Center, <<http://www.gordon.com/software-patents/statements/adobe.testimony.html>> (13 September 2006).

¹²⁹ Reichman has elsewhere observed that many of the major problems facing the patent system are due to a failure to protect these small grain-sized innovations. See Jerome H Reichman, “Of Green Tulips and Legal Kudzu: Repackaging Rights in Subpatentable Innovation” (2000) 53 *Vanderbilt Law Review* 1753.

¹³⁰ See Jerome H Reichman, “Of Green Tulips and Legal Kudzu: Repackaging Rights in Subpatentable Innovation” (2000) 53 *Vanderbilt Law Review* 1753, wherein he considers the market life of a hypothetical innovation in the plant-breeding industry, showing how the original innovator is adequately compensated for his contribution to the industry, whilst open access facilitates follow-on innovation. Reichman’s conclusion is that rather than just providing equivalent protection, some innovators are likely to be better off.

as they are for the lawyer and not the programmer, but the disclosure is in a familiar form, in code. That disclosure therefore provides access to material by which programmers can develop their evaluative aesthetic. In addition, it may well trigger the generational aesthetic, inspiring the creation of follow-on innovations.

Samuelson *et al* did not elaborate on the exact details of how such a system should work, preferring to focus on the framework of the system with a view to “facilitat[ing] and direct[ing] the political debate”.¹³¹ A question thus arises as to how to such a market-oriented system could be configured to meet the needs of the local industry. The first stage in such a process would be the gathering of empirical data as to the life cycle of software innovations. Such a process should be enhanced by a consultative process, gathering the views of all interested parties in the national software market. During the initial iteration, it may be advisable to adopt a conservative position, so that those holding software patents are not disadvantaged. Subsequent iterations could be informed by the reporting on information captured during the registration of innovations, which should be freely accessible. This data should be used to regularly (perhaps annually) assess and tweak system parameters, such as innovation shelf-life to match the actual life of innovations.¹³²

Davis et al, in introducing the paradigm in 1995, made the following remarks which are just as relevant today:

Each time there appears to be a lull in the controversy about legal protection for software, we are quickly jolted by the battle being joined anew. The difficulties wont soon disappear, ... because there is a deep seated problem here: existing intellectual property laws are fundamentally ill-suited to software.¹³³

The failure to adopt such a model means that the cycle continues anew. As mentioned in the Introduction, the next instalment of the patent wars has started, the battleground this time being smartphones.¹³⁴ The battle in Europe is just about to resume after a 5 year hiatus, this time over the unitary patent regime. If the software industry is to have any certainty, surely it is time to correct the mistakes of the past. It seems clear that patent law cannot accommodate software’s unique nature, and consequent needs. Therefore it is time to move forward and consider how the landscape without software patents might look. This alternative regime provides a glimpse of that future.

¹³¹ Pamela Samuelson et al, “A Manifesto Concerning the Legal Protection of Computer Programs,” (1994) 94 *Columbia Law Review* 2308 at 2315.

¹³² On the operation of a similar regime for the protection of copyright, see William W Fisher III, *Promises to Keep: Technology, Law and the Future of Entertainment* (Stanford University Press, 2004).

¹³³ Randall Davis et al, “A New View of Intellectual Property and Software” (1996) 39 *Communications of the ACM* 21 at 21.

¹³⁴ See Introduction, n 22 on page 4 and accompanying text.

Conclusion

The primary object of this thesis was to examine the patentability of software, and in particular to put forward the reasons why software should not be patentable. The argument proceeded from some observations in Chapter 1 about the history of software, and the way in which it was developed. It was seen from the history of software how successive layers of abstraction away from the physical implementation of the computer formed the foundation for new developments, and also allowed programmers to tackle problems of an ever-increasing increase in complexity. The development of software was also demonstrated to be a journey away from the ultimate in abstract artefacts, the idea, towards a (somewhat) less abstract implementation of that idea in software, travelling through various levels of decreasing abstractness towards that goal. The chapter then explored the assertion that mathematics and software are identical, or isomorphic. It was shown that this assertion was substantiated by its historical origins in the logicist and formalist programmes in mathematics. It was also verified by a correspondence between the practice of mathematics, and the practice of software development.

In Chapter 2, it was seen how the scope of patentable subject matter developed, as the coal face of technology moved forward, from entire trades and devices, through to more abstract subject matter. As the borderlines moved outward, courts developed new analytical tools to capture different aspects of the boundaries of inherent patentability, which were more useful in some instances than in others. However, amongst all of this change, two features remained constant. Firstly, the development of new analyses did not displace the underlying scope of patentability. Secondly, the nature of invention to which patent law has always extended, is limited to physical inventions.

Chapter 3 charted the patentability of software, from excluded abstract principle, or mental process, to its recharacterisation as either a physical artefact, or a process having a physical effect. Despite a brief period around the turn of the century when it looked like the physicality requirement might be thrown away, recent developments have made clear that physicality remains an important limit on the scope of patentability. The chapter demonstrated both that software and software-related inventions sit right at the borderline of patentable subject matter and, that defining the boundaries of patentability is a very difficult task.

Conclusion

Despite this inclusion of software within the bounds of patent law, in Chapter 4 it was shown how the patenting of software was hard to justify, both as a practical matter for those developing software, and also as a matter of theory. The way in which courts and tribunals in every jurisdiction have been confounded in their attempts to put forward a compelling explanation of why it is that software should or should not be patentable lead to the conclusion that a new approach is required.

The development of just such a new approach began in Chapter 5, where the nature of mathematics, as understood by both mathematicians and patent lawyers was considered. One of the important lessons from the history of mathematics, and attempts to set out a theoretical foundation for the discipline, is that ontological accounts, at least in mathematics, are somewhat of a red herring. After countless centuries, the nature of mathematics has not been resolved. All accounts considered capture something of the essence of mathematics, but all have their problems.

An attempt was made nonetheless to reconcile the status of mathematics within patent law by reference to the traditional exclusions developed. This was done by taking a holistic view, in that an explanation of mathematics' non-patentability should only be accepted if it could be reconciled with all theories of mathematics. It was thereby demonstrated that no traditional exclusion gave a complete account of the non-patentability of mathematics.

Chapter 6 took a different approach, asking not what mathematics was, but what it required for its advancement. The answer to that question was much more fruitful, and lead to an investigation of the role of freedom in mathematics. From that explanation, and a consideration of the philosophy of technology, a three dimensional framework of analysis was developed.

From the application of that framework to software in Chapter 7, it was demonstrated why programming was not a useful art, a technology, but a fine art. As far as most of us are concerned, it is the symbolic aspects of software which are most important, not the physical. Whilst it is possible that a computable process might form part of a physical device, claims artefacts existing higher up the stack, such as algorithms and abstract data structures, are so abstract that they ought never to be considered patentable. It was shown that despite the fact that implements a function, the expressiveness of software permeates every level of software development, and every aspect of software even when in executable form. Finally, the logical sequence of steps, and the application of software to "serious" projects from science to business, as well as the desire of some to relive the early days of software and transform software into a form of engineering, the software development process does not follow a rational, sequential path like the idealised waterfall model suggests. Software is evaluated by, motivated by and generated by aesthetics. As such, programming is better described as a fine art than a useful art, and would, in the vast majority of cases at least, not be patentable subject matter.

Application of the analysis to software also satisfied a further aim of this thesis – to identify commonly misunderstood aspects of the nature of software. In particular, it was explained how software, which might on a cursory analysis appear to be physical, either by reference to its storage on physical media, or its effect of turning a computer into a new machine, is at its core an intangible artefact. Secondly, a consideration of the role of the aesthetic in evaluation,

motivation and generation of software was shown to belie the rigorous, logical sequence of steps in which it takes its final form. It is perhaps in this aspect in which the asserted identity between mathematics and software finds its strongest manifestation.

In Chapter 8, the consequences of that position were explored. As a starting point, it was shown how the framework developed gave due consideration to the important features of software noted in Chapter 1. It was then noted that each of the jurisdictions surveyed in the thesis gave at least some consideration to the physicality/intangibility dimension of the analysis, but only lesser consideration to the other two aspects.

The final aim stated at the outset, was to develop a mode of analysis which went beyond a narrow, technical analysis of patent law. It is clear that the framework developed satisfies that goal. Its genesis was an acknowledgment of the role of freedom in mathematics, and it therefore demonstrates how fundamental human rights concepts are at work in patent law, not as an external influence requiring recharacterisation, but as concepts which flow through the development of traditional patent law principles. It also demonstrates how novel investigative tools, directed to other relevant considerations, such as social, moral and ethical considerations might be developed in a similar manner.

More broadly, it was argued that the need to exclude software from patentability, and the distinction between fine arts and useful arts, suggested that the subject matter inquiry has continuing relevance as a gatekeeper in patent law. No amount of tinkering with the other technical tests, examination regime improvements, or private initiatives are likely to be successful. The role of the inherent patentability inquiry is unique within patent law, as the only way in which patent law can properly engage with policy, ethical and social issues upon which the patent system has an impact. The relationship of the framework to the freedoms of thought and expression demonstrates how it is possible to craft an analytical tool which is consistent with the history of patent law, but also allows room for these “soft” issues to be considered in a structured way.

Finally, the implications of software’s non-patentability on its protection were considered. Removing patents as a protection mechanism for software does not leave a gaping hole. There is evidence suggesting the free rider problem is not as large as the pro-software-patent lobby might suggest. A variety of protections beyond patent law are presently used to protect different aspects of software. Either alone or in combination, it is possible to imagine how remaining paradigms provide protection to software which is appropriate to its nature. This is not a theoretical claim, but one borne out of the history of software before the award of software patents.

To address any remaining concerns about the gap in protection, an alternative protection paradigm was discussed. This paradigm, based as it is on a contemporary analysis of factors likely to fulfil the needs of innovators and competitors is one which is much contoured to the needs of this Information Age creation.

Conclusion

A Concluding Comment

As computer hardware gets smaller, faster, more connected, and more reliable, the software which controls it grows becomes ever more present in our daily lives. New modes of interaction, once the stuff of science fiction, such as voice recognition, motion sensing, face recognition and increased connectivity change the way we access information, connect with each other, work together, and play together. New advances in software have a central role to play in this context, and as such software undoubtedly has great promise. But it flows from both the reach of software, and its potential benefits, that we must consider carefully how best to encourage its advancement. Patent law is but one potential mechanism by which innovation might be encouraged. It has a long history, but it reflects certain policy choices about a balance between freedom and control which are connected to the types of physical inventions which have been within the historical scope of patent law. As software does not fit within that category, it ought not be patentable, and a new paradigm found to address its needs.

Bibliography

Articles/Books/Reports

A

Harold Abelson and Gerald Jay Sussman, *Structure and Interpretation of Computer Programs* (2nd ed, MIT Press, 1996)

Advisory Council on Intellectual Property, *Patenting of Business Systems* (Issues Paper, July 2002)

Advisory Council on Intellectual Property, *Report on a Review of the Patenting of Business Systems* (September 2003)

Advisory Council on Intellectual Property, *Patents and Experimental Use* (Options Paper, February 2004)

Advisory Council on Intellectual Property, *Patentable Subject Matter* (Issues Paper, July 2008)

Advisory Council on Intellectual Property, *Patentable Subject Matter* (Final Report, December 2010)

Evandro Agazzi, "The Rise of the Foundational Research in Mathematics," (1974) 27 *Synthese* 7

Chloe Albanesius, "Infographic: Who Will Win the Patent Wars?" *PCMag.com*, 2 September 2011 <<http://www.pcmag.com/article2/0,2817,2392375,00.asp>> (4 September 2011)

Christopher Alexander et al, *A Pattern Language: Towns, Buildings, Construction* (Oxford University Press, 1977)

John R Allison and Mark A Lemley, "Who's Patenting What? An Empirical Exploration of Patent Prosecution" (2000) 58 *Vanderbilt Law Review* 2099

WS Anglin, *Mathematics, a concise history and philosophy* (Springer Verlag, 1994)

Eugenio Archontopoulos et al, "When small is beautiful: measuring the evolution and consequences of the voluminosity of patent applications at the EPO" (2007) 19(2) *Information Economics and Policy* 103

Bibliography

Aristotle, *Nichomachean Ethics*, (Sir David Ross (trans), Oxford University Press, 1966)

Aristotle, *The Metaphysics of Aristotle* (HG Bohn, 1857)

Asish Arora and Alfonso Gambardella, "The Globalization of the Software Industry: Perspectives and Opportunities For Developed and Developing Countries" (National Bureau of Economic Research Working Paper 10538, June 2004) <<http://www.nber.org/papers/w10538>> (5 September 2011)

Michael F Atiyah, "The Evolution of Mathematics in the 20th Century," (2001) 108(7) *The American Mathematical Monthly* 654

Patrick Atiyah, "Book Review: Policy Arguments in Judicial Decisions by John Bell," (1985) 33(2) *American Journal of Comparative Law* 342

Daniel J M Attridge, "Challenging Claims! Patenting Computer Programs in Europe and the USA," (2001) 1 *Intellectual Property Quarterly* 22

Robert Audi (ed), *The Cambridge Dictionary of Philosophy* (2nd ed, Cambridge University Press, 1999)

Australian Patent Office, *Guidelines for Considering the Patentability of Computer Program Related Inventions* (Canberra, March 1986)

Australian Patent Office, *Manual of Practice and Procedure* (2006)

<http://www.ipaustralia.gov.au/pdfs/patentsmanual/WebHelp/Patent_Examiners_Manual.htm> (10 September 2011)

Jeremy Avigad and Erich H Reck, "Clarifying the nature of the infinite: the development of metamathematics and proof theory," (Carnegie Mellon Technical Report No CMU-PHIL-120, 11 December 2001)

B

Charles Babcock and Larry Greenemeier, "Open Source Stress" *Information Week*, 9 August 2004 <<http://www.informationweek.com/story/showArticle.jhtml?articleID=26806464>> (18 September 2010)

Walter William Rouse Ball, *A Short Account of the History of Mathematics*, (4th ed, Project Gutenberg, 2010)

Henk Barendregt and Freek Wiedijk, "The Challenge of Computer Mathematics" (2005) 363 *Philosophical Transactions: Mathematical, Physical and Engineering Sciences* 2351.

David Barker-Plummer "Turing Machines" *Stanford Encyclopedia of Philosophy* <<http://plato.stanford.edu/entries/turing-machine/>> (19 July 2008)

John D Barrow, *Pi in the Sky: Counting, Thinking, and Being* (Clarendon Press, 1992)

Andrew Beckerman-Rodau, "The Choice Between Patent Protection and Trade Secret Protection: A Legal and Business Decision," (2002) 84 *Journal of the Patent and Trademark Office Society* 371

- Paul Benacerraf and Hilary Putnam, *Philosophy of Mathematics* (2nd ed, Cambridge University Press, 1983)
- Amir M Ben-Amram, "The Church-Turing thesis and its look-alikes," (2005) 36(3) *SIGACT News* 113
- Yochai Benkler, "Free as the Air to Common Use: First Amendment Constraints on Enclosure of the Public Domain," (1999) 74 *New York University Law Review* 354
- Hugh H Benson (ed), *A Companion to Plato* (Blackwell Publishing, 2005)
- Jeremy Bentham, *An Introduction to the Principles of Morals and Legislation* (1996)
<<http://www.econlib.org/library/Bentham/bnthPML.html>> (1 July 2008)
- Edmund Berkeley, *Giant Brains, or Machines that Think* (1949)
- George Berkeley, *Treatise Concerning the Principles of Human Knowledge* (Project Gutenberg, 2009)
<<http://www.gutenberg.org/files/4723/4723-h/4723-h.htm>> (29 November 2010)
- William P Berlinghoff, *Mathematics: The Art of Reason* (DC Heath & Company, 1968)
- G Berrisford and M Burrows, "Reconciling OO with Turing machines" (1994) 37(10) *The Computer Journal* 888
- Daniel M Berry, "The Inevitable Pain of Software Development: Why There Is No Silver Bullet" in Martin Wirsing, Alexander Knapp and Simonetta Balsamo (eds), *Radical Innovations of Software and Systems Engineering in the Future, Proceedings of the 2002 Monterey Conference*, LNCS 2941 (Springer, 2004) 50
- James E Bessen and Robert M Hunt, "An Empirical Look at Software Patents" (2007) 16(1) *Journal of Economics & Management Strategy* 157
- James E Bessen and Eric Maskin, *Sequential Innovation, Patents, And Imitation* (2000) (Working Paper No 00-01, MIT Department of Economics) <<http://ssrn.com/paper=206189>> (29 June 2011)
- James E Bessen and Michael J Meurer, *Patent Failure: how judges, bureaucrats, and lawyers put innovators at risk* (Princeton University Press, 2008)
- Meurig Beynon, Russell Boyatt and Zhan En Chan, "Intuition in Software Development Revisited" in Jim Buckley, John Rooksby and Roman Bednarik (eds) *Proceedings of 20th Annual Psychology of Programming Interest Group Conference* (2008)
<<http://www.cs.st-andrews.ac.uk/~jr/papers/ppig08Proceedings.pdf>> (16 June 2011)
- David S Bir, "The Patentability Of Computer Software After Alappat: Celebrated Transformation or Status Quo?" (1995) 41 *Wayne L Rev* 1531
- Michael Birnhack, "More or Better? Shaping the Public Domain," in P Bernt Hugenholtz and Lucie Guibalt (eds), *The Future of the Public Domain* (Kluwer International, 2006) 59
- Barry Boehm, "A View of 20th and 21st Century Software Engineering" (Paper presented at the 28th International Conference on Software Engineering, Shanghai, China, 20 May 2006) 12
<<http://portal.acm.org/citation.cfm?id=1134288>>

Bibliography

- Klaus Boehm, *The British Patent System, Volume 1: Administration* (1967) extracted in Sam Rick-
etson and Miranda Richardson, *Intellectual Property: Cases, Materials and Commentary* (3rd ed,
LexisNexis 2005)
- George Boole, *An Investigation of the Laws of Thought* (Project Gutenberg, 2005)
- Luigi Borzacchini, *Being and Sign II: Axiomatic deductive method and infinite in Greek mathematics*
(1995) <<http://www.dm.uniba.it/~psiche/bas2/bas2.html>> (2 November 2010)
- Kathy Bowrey, Michael Handler and Dianne Nicol, *Australian Intellectual Property: Commentary,
Law and Practice* (Oxford University Press, 2010)
- James Boyle, *Shamans, software, and spleens : law and the construction of the information society*
(Harvard University Press, 1997)
- James Boyle, "Enclosing the Genome: What the Squabbles over Genetic Patents Could Teach
Us" in F Scott Kieff, *Perspectives on the Human Genome Project* (Academic Press, 2003) 97
- James Boyle, "The Second Enclosure Movement and the Construction of the Public Domain"
(2003) 66 *Law and Contemporary Problems* 33
- James Boyle, "A Manifesto on WIPO and the Future of Intellectual Property Protection," (2004)
2004 *Duke Law & Technology Review* 9
- Peter Bradley, "Turing Machines and Natural Language" *Consortium on Cognitive Science In-
struction*, 2002
<http://www.mind.ilstu.edu/curriculum/turing_machines/turing_machines_and_language.php>
(2 August 2010)
- Robert Bray, "The European Union 'Software Patents' Directive: What Is It? Why Is It? Where
Are We Now?" [2005] *Duke Law and Technology Review* 0011
- David Brennan, "The Evolution of English Patent Claims as Property Definers" [2005] 4 *Intel-
lectual Property Quarterly* 361; University of Melbourne Legal Studies Research Paper No. 185
<http://papers.ssrn.com/sol3/papers.cfm?abstract_id=938634> (17 December 2009)
- Frederick P Brooks Jr, "No Silver Bullet: Essence and Accidents of Software Engineering" (1987)
20(4) *Computer* 10
- Frederick P Brooks Jr, *The Mythical Man Month: Essays on Software Engineering* (Anniversary ed,
Addison-Wesley, 2002)
- James R Brown, *Philosophy of Mathematics: An Introduction to the World of Proofs and Pictures*
(Routledge, 1999)
- Joseph R Brown, "Software Patent Dynamics: Software As Patentable Subject Matter After State
Street & Trust Co.," (2000) 25 *Oklahoma City University Law Review* 639
- Dan L Burk and Mark A Lemley, "Is Patent Law Technology Specific?" (2002) 17 *Berkeley
Tech.Law Journal* 1155
- Dan L Burk and Mark A Lemley, "Policy Levers in Patent Law" (2003) 89 (7) *Virginia Law Review*
1575

Dan L Burk, "Patenting speech," (2000) 79 *Texas Law Review* 99

John A Burtis, "Towards a Rational Jurisprudence of Computer-Related Patentability in Light of *In re Alappat*," (1995) 79 *Minnesota Law Review* 1129

John B Bury, *A History of Freedom of Thought* (Williams and Norgate, 1914)

Samuel Butchart, *Evidence and Explanation in Mathematics* (PhD Thesis, Monash University, 2001)

Des Butler and Sharon Rodrick, *Australian Media Law* (2nd ed, Lawbook Co, 2004)

C

Florian Cajori, *A History of Mathematics* (Project Gutenberg, 2010)

Martin Campbell-Kelly, "Development and Structure of the International Software Industry, 1950-1990," (1995) 24(2) *Business and Economic History*

Martin Campbell-Kelly, "Not All Bad: An Historical Perspective on Software Patents" (2005) 11 *Michigan Telecommunications and Technology Law Review* 191

Martin Campbell-Kelly and Patrick Valduriez, "A Technical Critique of Fifty Software Patents," [2005] *SSRN eLibrary* <<http://ssrn.com/paper=650921>> (accessed 16 September 2008)

Michael A Carrier, "Unraveling the Patent-Antitrust Paradox" (2002) 150 *University of Pennsylvania Law Review* 761

Howard Caygill, *A Kant Dictionary* (Blackwell Reference, 1995)

Centre for Innovative Industry Economic Research Consortium, "Executive Summary", *The Australian software industry and vertical applications markets: Globally competitive and domestically undervalued*,

<http://www.dbcde.gov.au/__data/assets/pdf_file/0020/36461/Executive_summary.pdf> (5 September 2011)

Paul E Ceruzzi, *A History of Modern Computing* (MIT Press, 1998)

Siobhan Chapman, "A patently bad idea?" *Builder AU* (2004)

<<http://www.builderau.com.au/program/soa/A-patently-bad-idea-/0,339024614,339171460,00.htm>> (accessed 20 April 2007)

Donald S Chisum, "The Patentability of Algorithms," (1986) 47 *University of Pittsburgh Law Review* 959

Andrew Christie, "Designing Appropriate Protection for Computer Programs," (1994) 16(11) *European Intellectual Property Review* 486

Andrew Christie & Serena Syme, "Patents for Algorithms in Australia" (1998) 20 *Sydney Law Review* 517

Andrew Christie, "Some Observations on the Requirement of Inherent Patentability in the Context of Business Method Patents" (2000) 41 *IP Forum* 16

Bibliography

Anthony L Clapes, Patrick Lynch and Mark R Steinberg, "Silicon Epics and Binary Bards: Determining the Proper Scope of Copyright Protection for Computer Programs" (1987) 34 *UCLA Law Review* 1493

Roger Clarke, "A retrospective of the information systems discipline in Australia" in Guy G Gable (ed), *The Information Systems Academic Discipline in Australia* (1988) 75

M J Clugston et al (eds), *The New Penguin Dictionary of Science* (Penguin, 2004)

Julie E Cohen and Mark A Lemley, "Patent Scope and Innovation in the Software Industry," (2001) 89(1) *California Law Review* 3

Morris R Cohen, "Property and Sovereignty" (1927) 13 *Cornell Law Quarterly* 8

Sir Edmund Coke, *The third part of the Institutes of the Laws of England* (1644)

Timothy Colburn and Gary Shute, "Abstraction, Law and Freedom in Computer Science" (2010) 41(3) *Metaphilosophy* 345

Julian C Cole, "Mathematical Domains: Social Constructs?" in Bonnie Gold and Roger Simons (eds), *Proof and Other Dilemmas: Mathematics and Philosophy* (Mathematics Association of America, 2008)

Collins English Dictionary (Collins, 2000)

Mark Colyvan, "Indispensability Arguments in the Philosophy of Mathematics" in Edward N Zalta (ed) *The Stanford Encyclopedia of Philosophy* (Spring 2011 Edition)
<<http://plato.stanford.edu/archives/spr2011/entries/mathphil-indis/>> (28 June 2011)

B Jack Copeland, "The Church-Turing Thesis" *Stanford Encyclopedia of Philosophy*
<<http://plato.stanford.edu/entries/church-turing/>> (20 July 2008)

Robert I Coulter, "The Field of the Statutory Useful Arts: Part I" (1952) 34 *Journal of the Patent Office Society* 426

Robert I Coulter, "The Field of the Statutory Useful Arts: Part II" (1952) 34 *Journal of the Patent Office Society* 487

Robert I Coulter, "The Field of the Statutory Useful Arts: Part III" (1952) 34 *Journal of the Patent Office Society* 718

Brad Cox, "No Silver Bullet Revisited" *American Programmer Journal*, November 1995
<<http://virtualschool.edu/cox/pub/NoSilverBulletRevisited>> (22 July 2011)

Dennis Crouch, "Unreasonable Applicant Delay and the USPTO Backlog" *Patently-O*, 9 July 2010, <<http://www.patentlyo.com/patent/2010/07/unreasonable-patent-applicant-delay-and-the-uspto-backlog.html>> (12 July 2010)

Dennis Crouch and Robert P Merges, "Operating Efficiently Post-Bilski by Ordering Patent Doctrine Decision-Making" (2011) 25 *Berkeley Technology Law Journal* 1673

L Gordon Crovitz, "Patent Gridlock Suppresses Innovation" *Wall Street Journal*, 14 July 2008, <<http://online.wsj.com/article/SB121599469382949593.html>> (12 July 2010)

L Gordon Crovitz, "Google, Motorola and the Patent Wars" *The Wall Street Journal*, 22 August 2011 <<http://online.wsj.com/article/SB10001424053111903639404576518493092643006.html>> (4 September 2011)

Haskell Curry, "Functionality in Combinatory Logic" (1934) 20 *Proceedings of the National Academy of Sciences* 584

Haskell Curry and Robert Feys, *Combinatory Logic, Volume 1* (North-Holland, 1958)

Bill Curtis, "Fifteen Years of Psychology in Software Engineering: Individual Differences and Cognitive Science" ICSE '84 Proceedings of the 7th international conference on Software engineering (IEEE Press, 1984) <<http://portal.acm.org/citation.cfm?id=801956>> (27 July 2011)

Michael A Cusamano, *The Business of Software* (Simon and Schuster, 2004)

D

Arthur Danto, *The Abuse of Beauty: Aesthetics and the Concept of Art* (Open Court, 2003)

Harold L Davis, "Computer Programs and Subject Matter Patentability" (1977) 6 *Rutgers Journal of Computers and Law* 1

Martin Davis, *Computability and unsolvability* (Courier Dover Publications, 1982)

Philip J Davis and Reuben Hersh *Descartes' Dream; The World According to Mathematics* (Dover Publications, 1986)

Randall Davis et al, "A New View of Intellectual Property and Software" (1996) 39 *Communications of the ACM* 21

Anthony William Deller, "Book Review: Inventions, Patents and Monopolies by Meinhardt," (1947) 60(8) *Harvard Law Review* 1366

James DeLong, "Patent Reform Hits the Hill" *TechCentralStation.com*, 21 June 2005 <<http://techcentralstation.com/article.aspx?id=062105E>> (29 June 2011)

Department of Foreign Affairs and Trade, "Australia's reporting under ICESCR and ICCPR" <<http://www.dfat.gov.au/hr/reports/icescr-iccpr/index.html>> (accessed 8 April 2008).

René Descartes, *Meditations on First Philosophy* (Areté Press, 1986)

René Descartes, *Discourse on Method, Optics, Geometry, and Meteorology* (PJ Olscamp (trans), Hackett Publishing, 2001)

René Descartes, "Rules for the Direction of the Mind" in *Key Philosophical Writings* (Wordsworth Editions, 1997)

Scott Dexter et al, "On the Embodied Aesthetics of Code" (2011) 12 *Culture Machine* 1

Chris Di Bona and Sam Ockman (eds) *Open Sources: Voices from the Open Source Revolution* (O'Reilly & Associates, 1999)

Jean Dieudonné, "Abstraction in Mathematics and the Evolution of Algebra," in William E. Lamon (ed), *Learning and the Nature of Mathematics* (1972)

Bibliography

Edsger W Dijkstra, "The Humble Programmer", (1972) 15 (10) *Communications of the ACM* 859

Edsger W Dijkstra, "Programming as a Discipline of Mathematical Nature," (1974) 81(6) *The American Mathematical Monthly* 608

Peter Drahos, "Intellectual Property Law & Basic Science: Extinguishing Prometheus?" (1999) 10 *Law in Context* 56

Peter Drahos with John Braithwaite, *Information Feudalism: Who Owns the Knowledge Economy?* (Earthscan Publications, 2002)

Jay Dratler Jr, "Chapter 2: Obtaining Patent Rights" in *Intellectual Property Law: Commercial, Creative and Industrial Property* (1991)

Jay Dratler Jr, "Does Lord Darcy Yet Live? The Case Against Software and Business-Method Patents," (2003) 43 *Santa Clara Law Review* 823

Rochelle Dreyfuss, "Are Business Methods Patents Bad for Business?" (2000) 16 *Santa Clara Computer & High Technology Law Journal* 263

Rochelle Dreyfuss, "Patents and Human Rights: Where is the Paradox" in William Grosheide (ed), *Intellectual Property and Human Rights: A Paradox*, (Edward Elgar, 2010) 72

Peter F Drucker, *Technology, Management and Society* (Pan Books, 1970)

S B Drury, "Locke and Nozick on Property" (1982) 30(1) *Political Studies* 28

Michael A Dryja, "Looking to the Changing Nature of Software Development for Clues to Its Protection" (1995) 3 *University of Baltimore Intellectual Property Law Journal* 109

Gary Dukarich, "Patentability of Dedicated Information Processors and Infringement Protection of Inventions that Use Them" (1989) 29 *Jurimetrics Journal* 135

William W Dunham, *Euler: the master of us all* (The Mathematical Association of America, 1999)

Guy Waldo Dunnington, Jeremy Gray and Fritz-Egbert Dohse, *Carl Friedrich Gauss: titan of science* (MAA, 2004)

Alan L Durham, "'Useful Arts' in the Information Age" (1999) *Brigham Young University Law Review* 1419

Henry I Dutton, *The Patent System and Inventive Activity during the Industrial Revolution, 1750-1852* (1984)

E

Rebecca Eisenberg, "Analyze This: A Law and Economics Agenda for the Patent System" (2000) 53 *Vanderbilt Law Review* 2081

Paul Ernest, "Social Constructivism as a Philosophy of Mathematics: Radical Constructivism Rehabilitated?" (1990) <<http://people.exeter.ac.uk/PErnest/soccon.htm>> (10 September 2011)

Paul Ernest, *Social Constructivism as a Philosophy of Mathematics* (State University of New York Press, 1998)

Paul Ernest, "Is Mathematics Discovered or Invented?" (1999) 12 *Philosophy of Mathematics Education Journal* <<http://www.people.ex.ac.uk/PErnest/pome12/article2.htm>> (accessed 6 January 2008)

Andrei P Ershov, "Aesthetics and the Human Factor in Programming" (1972) 15(7) *Communications of the ACM* 501

European Economic and Social Committee, "Opinion of the Economic and Social Committee on the 'Proposal for a Directive of the European Parliament and of the Council on the patentability of computer-implemented inventions'" COM(2002) 92 final — 2002/0047 (COD) (19 September 2002)

European Patent Office, *Guidelines for Examination in the EPO* (1978), text reproduced at <<http://eupat.ffii.org/papri/epo-gl78/index.en.html>> (accessed 3 May 2010)

David S Evans and Anne Layne-Farrar, "Software Patents and Open Source: The Battle Over Intellectual Property Rights," (2004) 9 *Virginia Journal of Law & Technology* 10

William Bragg Ewald, *From Kant to Hilbert* (Oxford University Press, 1999) Volume 1

F

Robin Feldman, "The Open Source Biotechnology Movement: Is It Patent Misuse?" (2004) 6(1) *Minnesota Journal of Law, Science & Technology* 117

Joseph Feller et al (eds), *Perspectives on Free and Open Source Software* (MIT Press, 2005)

Frederick Ferré, *Philosophy of Technology* (Prentice Hall, 1988)

Hartry H Field, *Science Without Numbers: A Defence of Nominalism*, (Blackwell, 1980)

Jeremy Finn, "Particularism versus Uniformity: Factors Shaping the Development of Australasian Intellectual Property Law in the Nineteenth Century" (2000) 5 *Australian Journal of Legal History* 113

Alice E Fischer and Frances S Grodzinsky, *The Anatomy of Programming Languages* 1 (1993)

William W Fisher III, "Intellectual Property and Innovation: Theoretical, Empirical, and Historical Perspectives" (2001) 37 *Industrial Property, Innovation, and the Knowledge-based Economy*, *Beleidsstudies Technologie Economie*

Laura R Ford, "Alchemy and Patentability: Technology, 'Useful Arts' and the Chimerical Mind-Machine" (2005) 42 *California Western Law Review* 49

Robert Fortner, "Rest in Peas: The Unrecognized Death of Speech Recognition" <<http://robertfortner.posterous.com/the-unrecognized-death-of-speech-recognition>> (2 August 2010)

Harold Fox, *Monopolies and patents : a study of the history and future of the patent monopoly* (University of Toronto Press, 1947)

Bibliography

Maarten Franssen, Gert-Jan Lokhorst and Ibo van den Poel, "Philosophy of Technology," *Stanford Encyclopedia of Philosophy*, <<http://plato.stanford.edu/entries/technology/>> (1 May 2011)

Gottlob Frege, "The Thought: A Logical Inquiry" (1956) 65 *Mind* 289

Gottlob Frege, *The Basic Laws of Arithmetic* (Montgomery Furth (trans), University of California, 1967)

G

Denis Gallagher, "Smartphone operating systems review" *Choice Online*, 30 August 2011
<<http://bit.ly/qk646R>> (14 September 2011)

Erich Gamma et al, *Design Patterns: Elements of Reusable Object-Oriented Software* (Addison-Wesley Professional, 1994)

Michael C Gemignani, "Legal Protection for Computer Software: The View From '79" (1979-1980) 7 *Rutgers Journal of Computers Technology and Law* 269

Donald Gillies, "An Empiricist Philosophy of Mathematics and Its Implications for the History of Mathematics," in Emily Grosholz and Herbert Breger (eds), *The Growth of Mathematical Knowledge* (Springer, 2000) at 41-57

Murray Gleeson, "Judicial legitimacy" (2000) 20 *Australian Bar Review* 1

Kurt Gödel, *On Formally Undecidable Propositions of Principia Mathematica and Related Systems* (B Melzer (trans), Dover Publications, 1992)

Kurt Gödel, "Russell's Mathematical Logic" in 1944 in Paul Benacerraf and Hilary Putnam, *Philosophy of Mathematics* (2nd ed, Cambridge University Press, 1983)

Bonnie Gold and Roger Simons (eds), *Proof and Other Dilemmas: Mathematics and Philosophy* (Mathematics Association of America, 2008)

Richard Gold, "The Reach of Patent Law and Institutional Competence," (2003) 1 *University of Ottawa Law and Technology Journal* 263

Dina Q Goldin and Peter Wegner "The Church-Turing Thesis: Breaking the Myth" in *Proceedings of CiE 2005* 152

R L Goodstein, "Empiricism in mathematics," (1969) 23(1) *Dialectica* 50

Bruce Gordon, "A Lacuna in Cyberspace: Implied Terms in Consumer Contracts," (2000) 14(4) *Commercial Law Quarterly* 3

Timothy Gowers et al (eds), *The Princeton Companion to Mathematics* (Princeton University Press, 2008)

Edward Grant, *The foundations of modern science in the Middle Ages: their religious, institutional and intellectual contexts* (1996)

Scott Grattan, "Judicial Reasoning and the Adjudication of Airspace Trespass," (1996) 4 *Australian Property Law Journal* 13

John M Griem Jr, "Against a Sui Generis System of Intellectual Property for Software," (1993) 22 *Hofstra Law Review* 145

Clive Gringras, "UK Patents for Software-Related Inventions," (1996) 2(5) *Computer and Telecommunications Law Review* 208

H

S H H, "Patenting A Principle" (1868) 16(3) *The American Law Register* 129

Ian Hacking, *The Social Construction of What?* (Harvard University Press, 1999)

Marci A Hamilton, "Power, Responsibility, and Republican Democracy" (1995) 93 *Michigan Law Review* 1539

Godfrey H Hardy, *A Mathematician's Apology* (Cambridge University Press, 1992)

Rom Harre, "Computation and the mind," (1997) 40(6) *The American Behavioral Scientist* 685

John R Harrison, "Chapter 3 - Lambda calculus as a programming language," in *Introduction to Functional Programming* (1997) <<http://www.cl.cam.ac.uk/teaching/Lectures/funprog-jrh-1996/>> (accessed 4 November 2008)

Jeff Hecht, "Boy takes swing at US Patents" *NewScientist*, 17 April 2002, <<http://www.newscientist.com/article/dn2178-boy-takes-swing-at-us-patents.html>> (12 July 2010)

Georg WF Hegel, *Philosophy of Right* (TM Knox (trans), Clarendon Press, 1952) <<http://www.marxists.org/reference/archive/hegel/works/pr/>> (accessed 12 April 2008)

Joachim Henkel and Mark Tins, "Munich/MIT Survey: Development of Embedded Linux" (Institute for Innovation Research, Technology Management and Entrepreneurship, University of Munich, 10 May 2004) <http://www.linuxfordevices.com/files/misc/MunichMIT-Survey_Embedded_Linux.pdf> (5 September 2011)

Reuben Hersh, *What is Mathematics, Really?* (Oxford University Press, 1997)

Reuben Hersh (ed), *18 Unconventional Essays on the Nature of Mathematics* (Springer, 2006)

Arund Heyting, "Intuitionistic views on the nature of mathematics," (1974) 27(1) *Synthese* 79

Jan Hilgevoord and Jos Uffink "The Uncertainty Principle" *Stanford Encyclopedia of Philosophy* <<http://plato.stanford.edu/entries/qt-uncertainty/>> (21 October 2008)

Eric D Hirsch, Joseph F Kett and James S. Trefil (eds), *The New Dictionary of Cultural Literacy* (Houghton Mifflin, 2002)

C A R Hoare, "The Emperor's Old Clothes" 1980 ACM Turing Award Lecture, (1981) 24(2) *Communications of the ACM* 75

C A R Hoare, "An axiomatic basis for computer programming" (1969) 12(10) *Communications of the ACM* 576

Bibliography

- Douglas R Hofstadter, *Gödel, Escher, Bach: An Eternal Golden Braid* (Basic Books, 1980)
- Lancelot Hogben, *Mathematics for the Million* (4th ed, Allen & Unwin, 1967)
- Lee A Hollaar, "Chapter 5: Software-Based Inventions" in *Legal Protection of Digital Information* (2002) <<http://digital-law-online.info/lpdi1.0/treatise67.html>> (20 August 2010)
- Andrew Joseph Hollander, "Patenting Computer Data Structures: The Ghost, The Machine and the Federal Circuit," [2003] *Duke Law & Technology Review* 0033
- Oliver W Holmes Jr, *The Common Law and Other Writings* (Little, Brown and Co, 1881)
- Leon Horsten, "Philosophy of Mathematics" *Stanford Encyclopedia of Philosophy*, 25 September 2007 <<http://plato.stanford.edu/entries/philosophy-mathematics>> (10 September 2011)
- William A Howard, "The formulae-as-types notion of construction", in Jonathan P Seldin and J Roger Hindley (eds), *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, (Academic Press, 1980) 479
- Jens Høyrup, *In measure, number, and weight: studies in mathematics and culture* (SUNY Press, 1994)
- Anton R Hughes, "A Comment on Software, Patents, Innovation and Openness" (2003) 14(1) *Journal of Law, Information and Science* 50
- Anton R Hughes, "Avoiding the Software Patent Problem: An Alternative Fix for TRIPS Junkies" (2007) 14(1) *eLaw: Murdoch University Electronic Journal of Law* 100
- Anton R Hughes, "Submission to the Advisory Council on Intellectual Property: Patentable Subject Matter Issues Paper" (19 September 2008)
<<http://www.acip.gov.au/reviewpatentable/IP22%20Anton%20Hughes.pdf>> (11 September 2011)
- Anton R Hughes, "Case Brief: Bilski v Kappos" (2010) 20 *Journal of Law, Information and Science* 229
- E Wyndham Hulme, "The History of the Patent System Under the Prerogative and at Common Law" (1896) 12 *Law Quarterly Review* 141
- E Wyndham Hulme, "The History of the Patent System Under the Prerogative and at Common Law – A Sequel" (1900) 16 *Law Quarterly Review* 44
- David Hume, *An Enquiry Concerning Human Understanding* (Project Gutenberg, 2006)
<<http://www.gutenberg.org/dirs/etext06/8echu10h.htm>> (29 November 2010)

I

- Industrial Property Advisory Committee, *Patents, Innovation and Competition in Australia* (Report, 29 August 1984)
- Industrial Property Advisory Council, *Patents, Innovation and Competition in Australia* (1984)

International Organization for Standardization, "Information Technology – Open Document Format for Office Applications (OpenDocument) v1.0" (ISO/IEC Standard No 26300:2006, 30 November 2006)

<http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=43485> (accessed 29 June 2011)

David Isles, "What evidence is there that 2^{65536} is a natural number?," (1992) 33(4) *Notre Dame Journal of Formal Logic* 465

J

Adam B Jaffe and Josh Lerner, *Innovation and its discontents: how our broken patent system is endangering innovation and progress and what to do about it* (2004)

Holman W Jenkins, "Obama and the Smartphone Wars" *The Wall Street Journal*, 24 August 2011 <<http://online.wsj.com/article/SB10001424053111903327904576526130093390612.html>> (4 September 2011)

Ralph E Johnson, "Frameworks = (components + patterns)," (1997) 40(10) *Communications of the ACM* 39

K

Brian Kahin, Matthew Shruers and Jonathon Band, "Brief for *Amicus Curiae* Computer & Communications Industry Association in Support of Petitioners" in *Ebay Inc and Half.com Inc v Mercexchange LLC* No 05-130 (2006)

Immanuel Kant, *Critique of Pure Reason* (Paul Guyer, Allen W Woods (trans, eds), Cambridge University Press, 1998)

David J Kappos, John R Thomas and Randall J Bluestone, "A Technological Contribution Requirement for Patentable Subject Matter: Supreme Court Precedent and Policy" (2008) 6(2) *Northwestern Journal of Technology and Intellectual Property* 152

Orin S Kerr, "Are We Overprotecting Code? Thoughts on First-Generation Internet Law," (2000) 57 *Washington & Lee Law Review* 1287

N Stephan Kinsella, "Against Intellectual Property," (2001) 15(2) *Journal of Libertarian Studies*

Ruloff F Kip Jr, "The Patentability of Natural Phenomena" (1952) 20 *George Washington Law Review* 371

Stephen C Kleene *Introduction to Metamathematics* (North-Holland, 1980)

Ben Klemens, "Software Patents Don't Compute," *IEEE Spectrum Online*, July 2005, <<http://spectrum.ieee.org/careers/careerstemplate.jsp?ArticleId=i070305>> (4 January 2008)

Ben Klemens, "New legal code: copyrights should replace software patents" (2005) *IEEE Spectrum Magazine*, <<http://spectrum.ieee.org/computing/software/new-legal-code>> (18 September 2010)

Bibliography

Morris Kline, *Mathematical Thought from Ancient to Modern Times* (Oxford University Press, 1972)

Andrew F Knight, "A Potentially New IP: Storyline Patents," (2004) 86(11) *Journal of the Patent and Trademark Office Society* <http://www.plotpatents.com/legal_analysis.htm> (accessed 5 December 2006)

Wilbur R Knorr, "Transcript of a Lecture Delivered at The Annual Convention of the History of Science Society, Atlanta, December 28, 1974" in Jean Christianidis (ed) *Classics in the history of Greek mathematics* (Kluwer Academic Publishers, 2004)

Donald E Knuth, "Structured Programming with go to Statements" (1974) 6(4) *Computing Surveys* 261

Donald E Knuth, "Computer Programming as an Art" (1974) 17(12) *Communications of the ACM* 667

Donald E Knuth, *Literate programming* (Center for the Study of Language and Information, Stanford University, 1992)

Gina Kolata, "Mathematicians Are Troubled by Claims on Their Recipes" *New York Times*, 12 March 1989

Hidetaka Kondoh, "What Is 'Mathematicalness' in Software Engineering?" in *Proceedings of Fundamental Approaches to Software Engineering 2000*, 163

Martin Kretschmer, "Editorial - JILT Issue 2003 (1)" (2005)
<http://www2.warwick.ac.uk/fac/soc/law/elj/jilt/2003_1/kretschmer> (11 February 2008)

L

James Lahore, "Computers and the Law: The Protection of Intellectual Property" (1978) 5 *Federal Law Review* 15

Gary Lea, "The Revolution That Never Was: A Cynic's Eye View of the Software, Business and E-Commerce Method Patenting Controversy in the Wake of State Street" (2000) 2(1) *Digital Technology Law Journal* 4

Peter Yun-hyoung Lee, "Inverting the Logic of Scientific Discovery: Applying Common Law Patentable Subject Matter Doctrine to Constrain Patents on Biotechnology Research Tools," (2005) 19(1) *Harvard Journal of Law and Technology*
<<http://jolt.law.harvard.edu/articles/pdf/v19/19HarvJLTech079.pdf>> (15 January 2008)

Rich Lehrbaum, "Linspire, Microsoft in Linux-related deal" *DesktopLinux.com*, 13 June 2007
<<http://desktoplinux.com/news/NS9642338710.html>> (4 September 2011)

Mark A Lemley and Eugene Volokh, "Freedom of Speech and Injunctions in Intellectual Property Cases" (1998) 48 *Duke LJ* 147

Mark A Lemley, "Property, Intellectual Property and Free Riding," (2005) 83 *Texas Law Review* 1031

Mark A Lemley et al, "Life After *Bilski*" (2011) 63 *Stanford Law Review* 1315

Thomas M Lennon and Shannon Dea, "Continental Rationalism" *Stanford Encyclopedia of Philosophy*, 21 November 2007 <<http://plato.stanford.edu/entries/continental-rationalism/>> (27 Sep 2010)

David Leonhardt, "John Tukey, 85, Statistician: Coined the Word 'Software'" *New York Times*, 28 July 2000, at A19

Øystein Linnebo, "Platonism in the Philosophy of Mathematics," *Stanford Encyclopedia of Philosophy*, 18 Jul 2009 <<http://plato.stanford.edu/entries/platonism-mathematics/>> (22 November 2010)

Bernard Linsky and Edward N Zalta, "What is Neologicism?" (2006) 12(1) *The Bulletin of Symbolic Logic* 60

John Littler, "Art and Computer Programming" *O'Reilly OnLamp.com*, 30 June 2005 <<http://onlamp.com/pub/a/onlamp/2005/06/30/artofprog.html>> (25 May 2011)

Ian Lloyd, "Intellectual Property in the Information Age," (2001) 23(6) *European Intellectual Property Review* 290

John Locke, *An Essay Concerning Human Understanding* (John W Yolton (ed), Dent, 1976)

Scott D Locke and William D. Schmidt, "Business Method Patents: The Challenge of Coping With an Ever Changing Standard of Patentability," (2008) 18 *Fordham Intellectual Property, Media and Entertainment Law Journal* 1079

Patricia L Loughlan, *Intellectual Property: Creative and Marketing Rights* (Lawbook, 1998)

Patricia L Loughlan, "Patents: Breaking Into the Loop" (1998) 20 *Sydney Law Review* 553

M

Bruce J MacLennan, "Aesthetics in Software Engineering" (Technical Report No UT-CS-06-579, Department of Electrical Engineering and Computer Science, University of Tennessee, 31 October 2006) <<http://web.eecs.utk.edu/~library/TechReports/2006/ut-cs-06-579.pdf>> (26 May 2011)

Christine MacLeod, *Inventing the Industrial Revolution: The English patent system, 1660-1800* (1988)

Christine MacLeod and Alessandro Nuvolari "Inventive Activities, Patents and Early Industrialization: A Synthesis of Research Issues" *Druid working paper 06-08*, Danish Research Unit for Industrial Dynamics, 2006

Penelope Maddy, *Realism in Mathematics* (Clarendon Press, 1990)

Penelope Maddy, "Indispensability and Practice" (1992) 89(6) *Journal of Philosophy* 275

Penelope Maddy, "Naturalism and Ontology" (1995) 3(3) *Philosophia Mathematica* 248

Penelope Maddy, *Naturalism in Mathematics* (Clarendon Press, 1997)

Bibliography

- Ben D Manevitz, "What's the Story with Storyline Patents - An Argument Against the Allowance of Proposed Storyline Patents and for the Rejection of Currently Pending Storyline Patent Applications," (2006) 24 *Cardozo Arts & Entertainment Law Journal* 717
- Penelope Maddy, "Physicalistic Platonism" in AD Irvine (ed), *Physicalism in Mathematics* (Kluwer, 1990)
- David Malament, "Review of Field's Science Without Numbers" (1982) 79(9) *Journal of Philosophy* 523
- Eli Maor, *To Infinity and Beyond: A Cultural History of the Infinite* (Birkhäuser, 1987)
- "March of the outdated machines" *New Scientist*, 21 September 2009
<<http://www.newscientist.com/gallery/dn17805-computer-museums-of-the-world/4>> (10 March 2010)
- George E Martin, *The foundations of geometry and the non-Euclidean plane* (Springer, 1998)
- Massachusetts Institute of Technology, "Archimedes' Death Ray Experiment" (2007)
<http://web.mit.edu/2.009/www/experiments/deathray/10_ArchimedesResult.html> (accessed 30 October 2007)
- John McCarthy "Towards a mathematical science of computation" in CM Popplewell (ed), *Information Processing 1962* (Holland Publishing Company, 1962) 21;
<<http://www-formal.stanford.edu/jmc/towards.html>> (21 July 2008)
- Norman D. McClaskey, "The Mental Process Doctrine: Its Origin, Legal Basis, and Scope" (1969) 55 *Iowa Law Review* 1148
- Robert E McGinn "What is Technology?" in Paul T. Durbin, (ed.) *Research in Philosophy and Technology*, Vol. I (JAI Press, 1978)
- Jill McKeough, Andrew Stewart and Phillip Griffith, *Intellectual Property in Australia* (3rd ed, LexisNexis Butterworths, 2004)
- Kenneth McLeish (ed), *Bloomsbury Guide to Human Thought* (Bloomsbury Publishing, 1993)
- Peter Meinhardt, *Inventions, Patents and Monopoly* (2nd ed, Stevens & Sons, 1950)
- Peter S Menell, "1600, Intellectual Property: General Theories," in *Encyclopedia of Law and Economics* (Boudewijn Bouckaert & Gerrit De Geest eds., 1996)
<<http://encyclo.findlaw.com/index.html>> (accessed 7 April 2008)
- Peter S Menell, "Intellectual Property and the Property Rights Movement," (2007) 30(3) *Regulation* <<http://ssrn.com/abstract=1000061>>
- Robert Merges "Property Rights for Business Concepts and Patent System Reform" (1999) 14 *Berkeley Technology Law Journal* 577
- Merriam-Webster's Dictionary of Law* (Merriam-Webster, 1996)
- Frank I Michelman, "Law's Republic" (1988) 97 *Yale Law Journal* 1493
- John Stuart Mill, *On Liberty* (eBooks@Adelaide, 2009)
<http://ebooks.adelaide.edu.au/m/mill/john_stuart/m645o/index.html> (28 June 2011)

John Stuart Mill, *A System of Logic Ratiocinative and Inductive being a Connected View of the Principles of Evidence and the Methods of Scientific Investigation*, (8th ed, Harper & Brothers, 1882)
<<http://www.gutenberg.org/files/27942/27942-pdf.pdf>> (10 September 2011)

John Stuart Mill, *Considerations on Representative Government* (eBooks@Adelaide, 2009)
<http://ebooks.adelaide.edu.au/m/mill/john_stuart/m645r/chapter2.html> (4 March 2011)

Oliver Mills, *Biotechnological Innovations: moral restraints and patent law* (2005)

Marvin L Minsky, *Computation: finite and infinite machines* (Prentice Hall, 1967)

Carl Mitcham, *Thinking through Technology: The Path Between Engineering and Philosophy* (University of Chicago Press, 1994)

Hazel VJ Moir, "Do Patent Systems Improve Economic Well-Being? An Exploration of the Inventiveness of Business Method Patents" September 2008,
<<http://thesis.anu.edu.au/public/adt-ANU20090621.135115/index.html>> (accessed 11 February 2010)

Peter Molzberger, "Aesthetics and Programming" in *Proceedings of the SIGCHI conference on Human Factors in Computing Systems* (1983) 247

Ann L Monotti, "The Scope Of 'Manner Of Manufacture' Under The Patents Act 1990 (Cth) After Grant v Commissioner Of Patents" (2006) 34(3) *Federal Law Review* 461

James H Moor, "Three Myths of Computer Science" (1978) 29 *British J Phil Sci* 213

Yiannis N Moshovakis, "On founding the theory of algorithms", in Harold G Dales and Gianluigi (eds), *Truth in Mathematics* (Oxford University Press, 1998) 71

N

Thomas B Nachbar, "Monopoly, Mercantilism, and the Politics of Regulation" (2005) 91(6) *Virginia Law Review* 1313

Ernest Nagel and James R Newman, *Gödel's proof* (NYU Press, 2001)

Ryan Naraine, "When Software Patents Go Bad" *Internetnews.com*, 13 September 2002,
<<http://www.internetnews.com/bus-news/article.php/1462741/When-Software-Patents-Go-Bad>> (12 July 2010)

Brigham Narins (ed), *World of Mathematics*, Volume 1 (Gale Group, 2001)

Sarah H Nash and Samuel T Redwine Jr, "People and Organizations in Software Production: A Review of the Literature" (1988) 11(3) *ACM SIGCPR Computer Personnel* 10

Peter Naur, "Intuition in Software Development" in Harmut Ehrig et al (eds) *Theory and Practice of Software Development*, Volume 2, LNCS 186 (Springer, 1985) 60

R David Nelson (ed), *The Penguin Dictionary of Mathematics* (Penguin, 2008)

Neil Weinstock Netanel, "Copyright and a Democratic Civil Society" (1996) 106 *Yale Law Journal* 283

Bibliography

Allen Newell, "Response: The Models Are Broken, The Models Are Broken" (1986) 47 *University of Pittsburgh Law Review* 1023

James R Newman (ed), *The world of mathematics* (Simon & Schuster, 1956)

James R Newman (ed), *The World of Mathematics, Volume 3* (Courier Dove Publications, 2000)

Tuan Nguyen, "Red Hat Says U.S. Patent System Is Broken" *Tom's Guide*, 8 April 2008, <<http://www.tomsguide.com/us/redhat-patent-lawsuit,news-961.html>> (12 July 2010)

Dianne Nicol, "On the Legality of Gene Patents," (2005) 29(3) *Melbourne University Law Review* 809

Robert O Nimtz, "The Patentability of Computer Programs" (1970) 1 *Rutgers Journal of Computers and Law* 38

NV Philips Gloeilampenfabrieken and Another v Mirabella International Pty Limited (1993) 44 FCR 239

O

John WL Ogilvie, "Defining Computer Program Parts Under Learned Hand's Abstractions Test in Software Copyright Infringement Cases" (1992) 91 *Michigan Law Review* 526

James Oldham, *The Mansfield Manuscripts and the Growth of English Law in the Eighteenth Century*, Volume 1 (1992)

David S Olson, "Taking the Utilitarian Basis for Patent Law Seriously: The Case For Restricting Patentable Subject Matter" (2009) 82 *Temple Law Review* 181

George Orwell, "Appendix: The Principles of Newspeak", *Nineteen Eighty Four*, <<http://www.netcharles.com/orwell/books/1984-Appendix.htm>> (16 April 2008)

Robert L Ostergard, "Intellectual Property: A Universal Human Right?" (1999) 21(1) *Human Rights Quarterly* 156

Oxford English Dictionary (2nd edition, 2009)

Effy Oz, "Acceptable Protection of Software Intellectual Property: A Survey of Software Developers and Lawyers." (1998) 34(3) *Information & Management* 161

P

David L Parnas and Paul C Clements, "A Rational Design Process: How and Why To Fake It" (1986) 12(2) *IEEE Transactions on Software Engineering* 251

David Pearce "Mewsings on the word 'technical'" *IPKat*, 19 December 2007 <<http://ipkitten.blogspot.com/2007/12/mewsings-on-word-technical.html>> (accessed 11 May 2010)

Leonard Peikoff, *Objectivism: The Philosophy of Ayn Rand* (Penguin Group, 1991)

- Michael Peters "Wittgenstein, Education and the Philosophy of Mathematics" (2002) 3 *Theory & Science* <<http://theoryandscience.icaap.org/content/vol003.002/peters.html>> (6 December 2010)
- Gale R Peterson, "Trade Secrets in an Information Age" (1995-1996) 32 *Houston Law Review* 385
- Ivars Peterson "Software's Origin" *Ivar Peterson's MathTrek Archive*, 31 July 2000
<http://www.maa.org/mathland/mathtrek_7_31_00.html> (10 March 2010)
- John R Pierce, *An Introduction to Information Theory* (1980)
- Jeremy Phillips, "Book Review: Patenting Software Under the European Patent Convention by Keith Beresford," (2000) 22(12) *European Intellectual Property Review* 605
- Justine Pila, "The Common Law Invention in its Original Form" [2001] *Intellectual Property Quarterly* 209
- Justine Pila, "Inherent patentability in Anglo-Australian law: a history," (2003) 14 *Australian Intellectual Property Journal* 109
- Justine Pila, "Dispute over the Meaning of 'Invention' in Article 52(2) EPC – The Patentability of Computer-Implemented Inventions in Europe," (2005) 36 *IIC: International Review of Intellectual Property and Competition Law* 173
- Justine Pila, "Article 52(2) of the Convention on the Grant of European Patents: What Did the Framers Intend? A Study of the Travaux Préparatoires," (2005) 36 *IIC: International Review of Intellectual Property and Competition Law*
<http://papers.ssrn.com/sol3/papers.cfm?abstract_id=736064> (11 February 2008)
- Justine Pila, "Response to the Australian Government Advisory Council on Intellectual Property's Request for Written Comments on Patentable Subject Matter" *Oxford Legal Studies Research Paper No. 37/2008*, September 19, 2008
<http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1274102> (accessed 30 November 2009)
- Justine Pila, "Software Patents, Separation of Powers, and Failed Syllogisms: A Cornucopia from the Enlarged Board of Appeal of the European Patent Office" (2010) *Oxford Legal Research Paper Series* 48/2010, (revised 1 June 2010)
- Plato, "The Republic" in *The Dialogues of Plato translated into English with Analyses and Introductions by B. Jowett, M.A. in Five Volumes* (3rd ed, Oxford University Press, 1892)
- Robert Plotkin, "Computer Programming and the Automation of Invention," (2003) 7(2) *UCLA Journal of Law and Technology*
<http://www.lawtechjournal.com/articles/2003/07_040127_plotkin.php> (accessed 19 April 2007)
- Karlis Podnieks, "Platonism, Intuitionism and the Philosophy of Mathematics" in (1992) 31 *Semiotika i informatika* 150, English translation available at
<<http://www.ltn.lv/~podnieks/gt1.html>> (29 Oct 2007)
- Jules Henri Poincaré, "Chapter 1: The Selection of Facts" *Science et méthode* (Francis Maitland (trans), 1908)

Bibliography

Richard A Posner, *Overcoming Law* (Harvard University Press, 1995)

Emil L Post, "Finite Combinatory Processes - Formulation 1" (1936) 1 *Journal of Symbolic Logic* 103

Presidential Commission on the Patent System, *To Promote the Progress of Useful Arts in an Age of Exploding Technology* (1966)

Hilary Putnam, "What is Mathematical Truth", in *Mathematics Matter and Method: Philosophical Papers*, Volume 1 (2nd ed, Cambridge University Press, 1979)

Helena M Pycior, "Mathematics and Philosophy: Wallis, Hobbes, Barrow, and Berkeley," (1987) 48(2) *Journal of the History of Ideas* 265

Q

WV Quine, "Two Dogmas of Empiricism" (1951) 60 *The Philosophical Review* 20

R

Anthony Ralston, Edwin D Reilly and David Hemmendinger (eds), *Encyclopedia of Computer Science* (4th ed, Wiley, 2003)

Matt Ratto, "Leveraging Software, Advocating Ideology: Free Software and Open Source" paper presented to the 29th Research Conference on Information, Communication and Internet Policy (2001) <<http://arxiv.org/abs/cs.CY/0109103>> (27 June 2011)

Dan Ravicher, "Position Paper: Mitigating Linux Patent Risk" *Open Source Risk Management*, 2 August, 2004 <http://www.osriskmanagement.com/pdf_articles/linuxpatentpaper.pdf> (18 September 2010)

Eric S Raymond, *The Cathedral and the Bazaar: musings on Linux and Open Source by an accidental revolutionary* (O'Reilly Media, 2001)

VJ Rayward-Smith, *A First Course in Computability* (Blackwell Scientific, 1986)

Michael Rebeiro, "Compton's Multimedia Patent: Should the Patent System be Used to Protect Software" (1995) 1(3) *Computer and Telecommunications Law Review* 81

Erich H Reck and Michael P Price, "Structures and Structuralism in Contemporary Philosophy of Mathematics" (2000) 125 *Synthese* 341

Jerome Reichman, "Legal Hybrids Between the Patent and Copyright Paradigms" (1994) 94 *Columbia Law Review* 2432

Jerome Reichman, "Of Green Tulips and Legal Kudzu: Repackaging Rights in Subpatentable Innovation" (2000) 53(6) *Vanderbilt Law Review* 1743

Report of the Committee appointed by the Attorney-General of the Commonwealth to Consider what Alterations are Desirable in the Patent Law of the Commonwealth (the Dean Committee), 1952

- Michael D Resnik, "Mathematics as a Science of Patterns: Ontology and Reference" (1981) 15 *Noûs* 529
- Michael D Resnik, "How Nominalist is Harry Field's Nominalism" (1985) 47(2) *Philosophical Studies* 163
- Michael D Resnik, *Mathematics as a science of patterns* (Clarendon Press, 1997)
- Rocque Reynolds and Natalie P Stoianoff, *Intellectual Property: Text and Essential Cases* (2nd ed, Federation Press, 2005)
- Sam Ricketson, "The Future Role of Australian Intellectual Property Law Reform and Administration" (1992) 3(1) *Australian Intellectual Property Journal* 3
- Sam Ricketson and Miranda Richardson, *Intellectual Property: Cases, Materials and Commentary* (3rd ed, LexisNexis 2005)
- Teresa Riordan, "The Patent Office faces huge backlogs, extremely technical inventions, and absurd ones" *New York Times*, 13 May 2002, <<http://nyti.ms/nQap4O+>> (12 July 2010)
- John Richards, "Patent Law Harmonization: A Historical Perspective" *Mondaq*, 22 June 2009 <<http://www.mondaq.com/unitedstates/article.asp?articleid=81474>> (13 July 2010)
- Michael Risch, "Everything is Patentable" (2008) 75 *Tennessee Law Review* 591
- Russ Roberts, "Why Linux is Wealthier than Microsoft" *BusinessWeek Online*, 19 November 2003 <http://yahoo.businessweek.com/technology/content/nov2003/tc20031119_9737.htm> (18 September 2010)
- Tony Robbin, *Shadows of reality: the fourth dimension in relativity, cubism, and modern thought* (Yale University Press, 2006)
- Alan Rosen and David Rosen, "The Turing Machine Revisited: The computational complexity of a visually conscious machine" <<http://www.mcon.org/submtd/turing.pdf>> (2 August 2010)
- Winston Royce, "Managing the Development of Large Software Systems," *Proceedings of Westcon* (IEEE CS Press, 1970) 328
- G Rozenberg and A Salomaa, "Formal languages and automata" in *Encyclopaedia of Mathematics* (Springer-Verlag, 2002) <<http://eom.springer.de/f/f040850.htm>> (accessed 28 June 2011)
- Bertrand Russell, "The Study of Mathematics" in Andrew Brink, Margaret Moran and Richard A Rempel (eds), *Collected Papers of Bertrand Russell, Volume 12* (Routledge, 1985) 83

S

- H Sackman, W J Erickson and E E Grant, "Exploratory Experimental Studies Comparing Online and Offline Programming Performance" (1968) 11(1) *Communications of the ACM* 3
- Jack L Sammons, "Censoring Samba: An Aesthetic Justification for the Protection of Speech," (2008) 37(3) *Stetson Law Review* 855

Bibliography

- Pamela Samuelson, "Benson Revisited: The Case Against Patent Protection for Algorithms and Other Computer Program-Related Inventions," (1990) 39 *Emory Law Journal* 1025
- Pamela Samuelson et al, "A Manifesto Concerning the Legal Protection of Computer Programs," (1994) 94 *Columbia Law Review* 2308
- Pamela Samuelson, "Foreword: The Digital Content Symposium" (1997) 12(1) *Berkeley Technology Law Journal*
<<http://www.law.berkeley.edu/journals/btlj/articles/vol12/Samuelson/html/reader.html>> (25 February 2008).
- Pamela Samuelson and Suzanne Scotchmer, "The Law and Economics of Reverse Engineering" (2002) 111 *Yale Law Journal* 1575
- Risto Sarvas, A software engineering view to the decisions of the European Patent Office concerning computer programs, October 10, 2001 <http://www.tkk.fi/~rsarvas/Risto_Sarvas_thesis.pdf>
- Kurt M Saunders, "Reforming the Patent System: Two Proposals" (2005) 4(1) *The Technology Report*
<<http://www4.ncsu.edu/~baumerdl/The%20Technology%20Report/Revised.Reforming.the.Patent.System.pdf>> (13 July 2010)
- Sara Schaeffer, "Patent No. 6,004,596: Peanut Butter and Jelly Sandwich" *Wall Street Journal*, 5 April 2005 <<http://online.wsj.com/article/0,,SB111266108673297874,00.html>> (12 July 2010)
- Jürgen Schmidhuber, "Low Complexity Art" (1997) 30(2) *Leonardo* 97
- Stephen J Schnably, "Property and Pragmatism: A Critique of Radin's Theory of Property and Personhood" (1993) 45(2) *Stanford Law Review* 347
- D Seaborne Davies, "Early History of the Patent Specification" (1934) 50 *LQR* 86
- Lisa Shabel, "Kant's Philosophy of Mathematics" in Paul Guyer (ed) *The Cambridge Companion to Kant and Modern Philosophy* (Cambridge University Press, 2006)
- Carl Shapiro, "Patent System Reform: Economic Analysis and Critique" (2004) 19 *Berkeley Technology Law Journal* 1017
- Stewart Shapiro, "Conservativeness and Incompleteness" (1983) 80(9) *Journal of Philosophy* 521
- Stewart Shapiro, *Philosophy of Mathematics: Structure and Ontology* (Oxford University Press, 1997)
- Stewart Shapiro, *Thinking About Mathematics* (Oxford University Press, 2000)
- Mary Shaw, "Prospects for an Engineering Discipline of Software" (1990) 7(6) *IEEE Software* 15
- Mary Shaw, "Continuing Prospects for an Engineering Discipline of Software" (2009) 26(6) *IEEE Software* 64
- Brad Sherman, "The patentability of computer-related inventions in the United Kingdom and the European Patent Office" (1991) 13(3) *European Intellectual Property Review* 85
- Brad Sherman and Lionel Bently, *The Making of Modern Intellectual Property Law: The British Experience 1760-1911* (1999)

- Martin L Shooman, *Reliability of computer systems and networks: fault tolerance, analysis and design* (John Wiley and Sons, 2002)
- Nathalie Sinclair, "Aesthetics as a liberating force in mathematics education?" (2009) 41 *ZDM* 45
- Nathalie Sinclair, "The Roles of the Aesthetic in Mathematical Inquiry" (2004) 6(3) *Mathematical Thinking and Learning* 261
- Nathalie Sinclair and William Higginson (eds) *Mathematics and the Aesthetic: new approaches to an ancient affinity* (Springer, 2007)
- Jean Paul Smets and Harmut Pilch, "Software Patentability with Compensatory Regulation: a Cost Evaluation, (2001) 2(6) *Upgrade* 23
<<http://www.upgrade-cepis.org/issues/2001/6/up2-6Smets.pdf>> (4 May 2010)
- Adam Smith, *An Inquiry into the Nature and Causes of the Wealth of Nations* (5th ed, Methuen, 1904) <<http://www.econlib.org/library/Smith/smWN.html>> (accessed 2 June 2011)
- David Eugene Smith, *History of Mathematics* (Courier Dover Publications, 1958)
- Elliot Sober, "Mathematics and Indispensability" (1993) 102(1) *Philosophical Review* 35
- Morten Heine B Sørensen and Pawel Urzyczyn, *Lectures on the Curry-Howard isomorphism* (1998) <<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.17.7385>> (accessed 28 June 2011)
- Peter G Spivack, "Comment: Does Form Follow Function? The Idea/Expression Dichotomy in Copyright Protection of Computer Software," (1988) 35 *UCLA Law Review* 723
- Stanislaw J Soltysinski, "Computer Programs and Patent Law: A Comparative Study" (1973-1974) 3 *Rutgers Journal of Computers and Law* 1
- Richard Stallman, "Patent absurdity" *The Guardian*, 20 June 2005
<<http://www.guardian.co.uk/technology/2005/jun/20/comment.comment>> (4 September 2005)
- Richard Stallman, "Beware: Europe's 'unitary patent' could mean unlimited software patents" *The Guardian*, 22 August 2011 <<http://www.guardian.co.uk/technology/2011/aug/22/european-unitary-patent-software-warning>> (11 September 2011)
- Jack Starke, "The Patenting of Animal Forms With New Traits," (1987) 61 *Australian Law Journal* 324
- Mark Stefik and Daniel G Bobrow, "Object-Oriented Programming: Themes and Variations," (1986) 6(4) *AI Magazine* 40
- Mark Steiner, "Mathematics Applied: The Case of Addition" in Bonnie Gold and Roger Simons (eds), *Proof and Other Dilemmas: Mathematics and Philosophy* (Mathematics Association of America, 2008) 313
- Richard H Stern, "Scope-Of-Protection Problems With Patents and Copyrights On Methods of Doing Business," (1999) 10 *Fordham Intellectual Property, Media and Entertainment Law Journal* 105

Bibliography

Sarah L Stirland, "Bush makes new push for patent office to keep fees" *GovExec.com*, 6 February 2006, <http://www.govexec.com/story_page.cfm?articleid=33317> (15 August 2011)

Gregory Stobbs, *Software Patents* (2nd ed, Aspen, 2000)

Natalie Stoianoff, "Patenting Computer Software: An Australian Perspective," [1999] *European Intellectual Property Review* 500

Donald E Stokes, *Pasteur's Quadrant - Basic Science and Technological Innovation* (Brookings Institution Press, 1997)

Randall Stross, "Why Bill Gates Wants 3,000 New Patents" *New York Times*, 31 July 2005, <<http://nyti.ms/hzgnGC>> (12 July 2010)

Dirk J Struik, *A Concise History of Mathematics*, (3rd ed, Dover Publications, 1967)

Elsbeth Summers and Andrew Holmes (eds), *Collins Australian Dictionary and Thesaurus* (3rd ed, HarperCollins Publishers, 2004)

Frank Swetz, *Capitalism and Arithmetic: The New Math of the 15th Century* (David E Smith (trans), Open Court Publishing, 1987)

T

Puay Tang, John Adams and Daniel Paré, "Patent protection of computer programmes" (Final Report submitted to the European Commission, 2001)

The Hutchinson Unabridged Encyclopedia including Atlas, (Helicon, 2005)

John R Thomas, "The Patenting of the Liberal Professions," 40 *Boston College Law Review* 1139

John R Thomas, "The Post-Industrial Patent System," (1999) 10(3) *Fordham Intellectual Property Media and Entertainment Law Journal* 3

Gregory S Tseytin "From Logicism to Proceduralism (An Autobiographical Account)" in *Algorithms in Modern Mathematics and Computer Science* (Springer Verlag: Berlin 1979) 390

Robert Tubbs, *What is a Number? Mathematical Concepts and Their Origins* (John Hopkins University Press, 2009)

Ilkka Tuomi, "The Lives and Death of Moore's Law" (2002) 7(11) *First Monday*
<<http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/1000/921>> (18 September 2010)

Alan M Turing, "On Computable Numbers, with an Application to the Entscheidungsproblem" (1937) 47 *Proceedings of the London Mathematical Society* 230

Alan M Turing "Intelligent Machinery" in B Jack Copeland (ed), *The essential Turing: seminal writings in computing, logic, philosophy* (Oxford University Press, 2004)

U

United Kingdom, *Report of the Commissioners appointed to the Working of the Law relating to Letters Patent for Inventions* (C5974, 1865)

United Kingdom Patent Office, "1. Patentable Inventions" *Manual of Patent Practice – Patents Act 1977*, 1 July 2010 <<http://www.ipo.gov.uk/practice-sec-001.pdf>> (13 July 2010)

United States Bureau of Labour Statistics, "Computer Software Engineers and Computer Programmers" in *Occupational Outlook Handbook, 2010-11 Edition* <<http://www.bls.gov/oco/ocos303.htm>> (3 September 2011).

United States Department of State, "Australia: Country Reports on Human Rights Practices - 2005" <<http://www.state.gov/g/drl/rls/hrrpt/2005/61601.htm>> (accessed 12 April 2008)

United States Patent and Trademark Office, "Examination Guidelines for Computer-Related Inventions: Final Version" (1996) 61(40) *Federal Register* 7478

United States Patent and Trademark Office, "Interim Guidelines for Determining Subject Matter Eligibility Claims in View of *Bilski v Kappos*" (2010) 75(143) *Federal Register* 43922 (27 July 2010)

United States Patent and Trademark Office, *Performance and Accountability Report for Fiscal Year 2005*, <<http://www.uspto.gov/web/offices/com/annual/2005/2005annualreport.pdf>> (12 July 2010)

United States Patent and Trademark Office, "2106.02 Mathematical Algorithms," *Manual of Patent Examining Procedure*, Eighth Edition, August 2001, Revision 5, August 2006 <http://www.uspto.gov/web/offices/pac/mpep/documents/2100_2106_02.htm> (8 September 2011)

William Uzgalis, "John Locke," *Stanford Encyclopedia of Philosophy*, 2 September 2001, <<http://plato.stanford.edu/entries/locke/>> (30 November 2010)

V

Mark van Atten, "Luitzen Egbertus Jan Brouwer" in Edward N Zalta (ed), *Stanford Encyclopaedia of Philosophy*, (Summer 2011 Edition) <<http://plato.stanford.edu/archives/sum2011/entries/brouwer/>> (10 September 2011)

William van Caenegem, "The Technicality Requirement, Patent Scope and Patentable Subject Matter in Australia" (2002) 13 *Australian Intellectual Property Journal* 41

William van Caenegem, *Intellectual Property* (2nd ed, LexisNexis Butterworths, 2006)

Jan van Leeuwen and Jiri Wiedermann, "The Turing Machine Paradigm in Contemporary Computing" in B Enquist and W Schmidt (eds), *Mathematics Unlimited - 2001 and Beyond* (Springer Verlag, 2000)

Bibliography

Jan van Leeuwen and Jiri Wiedermann, "Beyond the Turing Limit: Evolving Interactive Systems," in *SOFSEM 2001: Theory and Practice of Informatics* (2001) <http://dx.doi.org/10.1007/3-540-45627-9_8> (accessed 3 November 2008)

Hal R Varian, "Patent Protection Gone Awry" *New York Times*, 21 October 2004, <<http://www.nytimes.com/2004/10/21/business/21scene.html>> (12 July 2010)

Greg R Vetter, "The Collaborative Integrity of Open-Source Software," 2004 *Utah Law Review* 563

Susan Vineberg, "Confirmation and the Indispensability of Mathematics to Science" (1996) 63(3) *Philosophy of Science* S256

John von Neumann, *First Draft of a Report on the EDVAC* (Moore School of Electrical Engineering, University of Pennsylvania, 30th June 1945) <<http://qss.stanford.edu/~godfrey/vonNeumann/vnedvac.pdf>> (9 March 2010)

Lev S Vygotsky and Alex Kozulin, *Thought and Language* (2nd ed, MIT Press, 1986)

W

John Wadham et al, *Blackstone's Guide to the Human Rights Act 1998* (4th ed, Blackstone, 2007)

Allen B Wagner, "Patenting Computer Science: Are Computer Instruction Writings Patentable?," (1998) 17 *John Marshall Journal of Computer and Information Law* 5

R Polk Wagner, "Of Patents and Path Dependency: A Comment on Burk and Lemley" (2003) 18 *Berkeley Technology Law Journal* 1341

Jeremy Waldron, "Property" in Edward N Zalta (ed) *Stanford Encyclopaedia of Philosophy* (Winter 2010 Edition) <<http://plato.stanford.edu/archives/win2010/entries/property/>> (29 April 2011)

HN Walker, *Australian Patents: The Australian Patent System Explained* (1949)

Henry M Walker, *The tao of computing* (Jones and Bartlett Learning, 2004)

Edward C Walterscheid "The Early Evolution of the United States Patent Law: Antecedents (Part 1)" (1994) 76 *Journal of the Patent and Trademark Office Society* 77

Edward C Walterscheid "The Early Evolution of the United States Patent Law: Antecedents (Part 2)" (1994) 76 *Journal of the Patent and Trademark Office Society* 849

Edward C Walterscheid "The Early Evolution of the United States Patent Law: Antecedents (Part 3)" (1995) 77 *Journal of the Patent and Trademark Office Society* 771

Edward C Walterscheid "The Early Evolution of the United States Patent Law: Antecedents (Part 3 continued)" (1995) 77 *Journal of the Patent and Trademark Office Society* 847

Edward C Walterscheid "The Early Evolution of the United States Patent Law: Antecedents (Part 4)" (1996) 78 *Journal of the Patent and Trademark Office Society* 77

Edward C Walterscheid "The Early Evolution of the United States Patent Law: Antecedents (5 Part 1)" (1996) 78 *Journal of the Patent and Trademark Office Society* 615

Edward C Walterscheid "The Early Evolution of the United States Patent Law: Antecedents (5, Part II)" (1996) 78 *Journal of the Patent and Trademark Office Society* 665

Edward C Walterscheid "To Promote the Progress of Useful Arts: American Patent Law and Administration, 1787–1836 (Part 1)" (1997) 79 *Journal of the Patent and Trademark Office Society* 61

Edward C Walterscheid "To Promote the Progress of Useful Arts: American Patent Law and Administration, 1787–1836 (Part 2)" (1998) 80 *Journal of the Patent and Trademark Office Society* 37

Edward C Walterscheid "To Promote the Progress of Useful Arts: American Patent Law and Administration, 1787–1836 (Part 3)" (1997) 79 *Journal of the Patent and Trademark Office Society* 61

Edward C Walterscheid "Patents and Manufacturing in the Early Republic" (1998) 80 *Journal of the Patent and Trademark Office Society* 855

Larry L Wear et al, *Computers: An Introduction to Hardware and Software Design* (McGraw-Hill, 1991)

Kimberlee Weatherall, "Samsung Galaxy Tab vs Apple iPad: the tablet patent wars hit Australia" *The Conversation*, 3 August 2011 <<http://theconversation.edu.au/samsung-galaxy-tab-vs-apple-ipad-the-tablet-patent-wars-hit-australia-2660>> (4 September 2011)

David B Webber, "Intellectual Property in internet software," in Anne Fitzgerald et al (eds), *Going Digital 2000: Legal Issues for E-Commerce, Software and the Internet* (2nd ed, Prospect Media, 2000)

Dane Weber, "A Critique of Intellectual Property Rights"
<<http://dane.weber.org/concept/thesis.html>> (accessed 12 April 2008)

Peter Wegner and Eugene Eberbach, "New Models of Computation," 47(1) *The Computer Journal*

Lauren Weinstein, "Stop the Patent Process Madness" *Wired*, 21 October 2002,
<<http://www.wired.com/politics/law/news/2002/10/55831>> (12 July 2010)

David Wells, "Are these the most beautiful?" (1990) 12(3) *The Mathematical Intelligencer* 37

Keith Werhan, *Freedom of Speech: A Reference Guide to the United States Constitution* (Greenwood Publishing Group 2004)

Robin Whaite & Ari Laakkonen, "Analysis: The EPO Simplifies Software Patenting," (1999) 4 *Intellectual Property Quarterly* 487

Alfred North Whitehead, *Introduction to Mathematics* (Williams & Northgate, 1911)

Benjamin Whorf, "Science and Linguistics" (1940) 42 *Technology Review* 229

Raymond M Wilder, "The Nature of Modern Mathematics," in William E Lamon (ed), *Learning and the Nature of Mathematics* (Science Research Associates, 1972)

Peter Williams, "Patent threat to open source is limited" *vnunet.com*, 8 September 2004
<<http://www.pcw.co.uk/vnunet/news/2125807/patent-threat-open-source-limited>> (18 September 2010)

Bibliography

Ludwig Wittgenstein, *On Certainty* (G.E.M Anscombe and G.H. von Wright, (trans), Wiley-Blackwell, 1975)

Ludwig Wittgenstein, *Tractatus Logico-Philosophicus* (Cosimo, 2010)

Kenneth Wong and Phet Sayo, "Free/Open Source Software: A General Introduction," *International Open Source Network*, 2004 <<http://www.iosn.net/foss/foss-general-primer/>> (18 September 2010)

Christopher Wonnell, "Truth and the Marketplace of Ideas," (1986) 19 *University of California at Davis Law Review* 669

John Worsley and Joshua Drake, "Chapter 3: Understanding SQL," in *Practical PostgreSQL* (2001) <<http://www.faqs.org/docs/ppbook/c1164.htm>> (accessed 28 June 2011)

R George Wright, "Why Free Speech Cases Are as Hard (And as Easy) as They Are," (2001) 68 *Tennessee Law Review* 335

Z

Edward N Zalta, "Frege's Logic, Theorem and Foundations for Arithmetic," *Stanford Encyclopedia of Philosophy*, 13 April 2007, <<http://plato.stanford.edu/entries/frege-logic/>> (27 March 2010)

Donald J Zeyl, Daniel Devereux and Phillip Mitsis (eds), *Encyclopedia of Classical Philosophy* (Greenwood Press, 1997)

Paola Zizzi, "Poetry of a Logical Truth"
<http://www.thalesandfriends.org/en/papers/pdf/zizzi_paper.pdf> (16 June 2008)

Cases

A

Advanced Building Systems Pty Limited v Ramset Fasteners (Aust) Pty Limited (1998) 194 CLR 171

Aerotel Ltd. v. Telco Holdings Ltd. and in the Matter of Macrossan's Application [2006] EWCA Civ 1371; [2007] RPC 7

Alexander v Haley 460 F.Supp 40 (1978)

Apple v Franklin 714 F.2d 1240 (3rd circuit, 1983)

Arrhythmia Research Technology v Corazonix Corporation 958 F.2d 1053 (1992)

Association for Molecular Pathology et al v USPTO et al 9 Civ 4515 (US District Court, Southern District of New York, 2010)

Association for Molecular Pathology v USPTO Appeal No 2010-1406 (Fed Cir, 2011)

AT&T Corp v Excel Communications Inc 172 F.3d 1352 (1999)

Atari Games Corp and Tengen Inc v Nintendo of America Inc and Nintendo Co Ltd 975 F.2d 832 (1992)

B

Badger Co Inc's Application [1969] FSR 474

Bernstein v United States Department of Justice 176 F.3d 1132 (9th Circuit, 1999)

Bilski v Kappos 130 S. Ct. 3218 (2010)

BGH GRUR 1696, 672 "Rote Taube"

Bonito Boats, Inc v Thunder Craft Boats, Inc 489 U.S. 141 (1989)

Boulton & Watt v Bull (1795) 126 ER 651

British United Shoe Machinery Co Ltd v Simon Collier Ltd (1909) 26 RPC 21

Bruce Howes v The Great Lakes Press Corporation, Holt Manufacturing Company and Devries Brothers 679 F.2d 1023 (1982)

Burroughs Corporation (Perkin's) Application [1973] RPC 147

C

Cain v Universal Pictures Co 47 F.Supp 1013 (1942)

CCOM Pty Ltd v Jiejing Pty Ltd (1994) 51 FCR 260

Central Hudson Gas and Electric Corp v Public Service Commission of New York 447 U.S. 557 (1980)

Citibank Na v The Comptroller General of Patents [2006] EWHC 1676 (Ch) (09 June 2006)

CFPH LLC's Application [2005] EWHC 1589 (Pat)

Cochrane v Deener 94 US 780 (1876)

Commissioner of Patents v Microcell Ltd (1959) 102 CLR 232

Computer Associates International, Inc. v Altai, Inc. 982 F.2d 693 (1992)

Computer Edge Pty Ltd and Another v Apple Computer Inc and Another (1985) 161 CLR 171

Cooper's Application (1902) 19 RPC 53

Crane v Price (1842) 134 ER 239

CyberSource Corporation v Retail Decisions Inc Appeal No 2009-1358 (Fed Cir, 2011)

D

Dann v Johnston 425 US 219 (1976)

David Kahn Inc v Conway Stewart [1974] RPC 486

Debs v United States 249 US 211 (1919)

Diamond v Bradley 450 US 381 (1981)

Diamond v Chakrabarty 447 US 309 (1980)

Diamond v Diehr 450 US 175 (1981)

Bibliography

E

eBay Inc v MercExchange LLC 547 U. S. 388 (2006)
Elton and Leda Chemicals Ltd's Application [1957] RPC 267
ESP's Application [1945] RPC 486
Ex parte Bowman (2001) 61 USPQ 1669
Ex parte Egan 128 USPQ 23 (BNA, 1960)
Ex parte King 146 USPQ 590 (BNA, 1964)
Ex parte Lundgren Appeal No 2003-2088 (BPAI, 2005)
Ex parte Meinhardt [1907] Dec. Com. Pat 237
Ex parte Murray 9 USPQ.2d (BNA) 1820 (1988)
In re Naquin 398 F.2d 863 (CCPA, 1968)

F

Financial Control Associates v Equity Builders 760 F Supp 1103 (1992)
Fishburn's Application (1940) 57 RPC 245
Flood v Coe 31 F Supp 348 (1940)
Fujitsu Ltd's Application [1996] RPC 511 (Pat)
Fujitsu Ltd's Application [1997] RPC 608 (Court of Appeal)

G

G_0003/08 PRESIDENT'S REFERENCE/Computer Program Exclusion [2009] EPOR 9 (Enlarged Board of Appeal, 12 May 2010)
Gale's Patent Application [1991] RPC 305
Genentech Inc's Application (Human Growth Hormone) [1989] RPC 147
Gottschalk v Benson 409 US 63 (1973)
Graham v John Deere Co 383 US 1 (1966)
Grant v Commissioner of Patents (2005) 67 IPR 62
Grant v Commissioner of Patents (2006) 154 FCR 62
Greenewalt v Stanley Co of America 12 USPQ 122 (1931)
Griffin v Isaacs (1938) 12 ALJ 169
Griswold v Connecticut 381 US 479 (1965)

H

Hall v Jarvis (1822) 1 Web PC 100

Halliburton Energy Services v Smith International [2005] EWHC 1623 (Pat)

Halliburton Oil Well Cementing Co. v Walker 146 F.2d 817 (1944)

Harwood v Great Northern Railway Company (1865) 11 HLC 654, 11 ER 1488.

Hickton's Patent Syndicate v Patents and Machine Improvements Co Ltd (1909) 26 RPC 339

Hornblower v Boulton (1799) 8 TR 99; 101 ER 1285

I

IBM's Application (1966) 26 AOJP 2395

IBM v Commissioner of Patents (1991) 33 FCR 218

In re Abele 684 F.2d 902 (1982)

In re Abrams 89 USPQ (BNA) 266 (1951)

In re Alappat 33 F.3d 1526 (1994)

In re Beauregarde 53 F.3d 1583 (1995)

In re Benson 441 F.2d 682 (1971)

In re Bernhart 417 F.2d 1395 (1969)

In re Bilski 545 F.3d 943 (Fed Cir, 2008)

In re Bradley 600 F.2d 811 (1979)

In re de Castelet 562 F.2d 1236 (1977)

In re Deutsch 553 F.2d 689 (1977)

In re Freeman 573 F.2d 1243 (1978)

In re Gelnovatch 595 F.2d 38 (1979)

In re Grams 888 F.2d 839 (1989)

In re Iwahashi 888 F.2d 1374 (1989)

In re Johnston 502 F.2d 765 (1974)

In re Maucorps 609 F.2d 484 (1979)

In re Meyer 688 F.2d 789 (1982)

In re Middleton 167 F.2d 1012; 77 U.S.P.Q. (BNA) 615 (1948)

In re Musgrave 431 F.2d 882 (CCPA, 1970)

In re Pardo 684 F.2d 915 (1982)

Bibliography

In re Prater 415 F.2d 1378 (1968)
In re Prater (1969) 415 F.2d 1390
In re Russell 48 F.2d 1395 (1931)
In re Sarkar 588 F.2d 1330 (1978)
In re Schrader 22 F.3d 290 (1994)
In re Sherwood 618 F.2d 758 (1980)
In re Taner 681 F.2d 787 (1982)
In re Telefon A/B LM Ericsson's Application [1975] FSR 49; 44 AOJP 846
In re Toma 575 F.2d 872 (1980)
In re Trovato 60 F.3d 807 (1995)
In re Waldbaum 559 F.2d 611 (1977)
In re Walter 618 F.2d 766 (1980)
In re Warmerdam 33 F.3d 1354 (1994)
In re Yuan 188 F.2d 377 (1951)
Invention Pathways Pty Ltd [2010] APO 10 (21 July 2010)

J

Joos v Commissioner of Patents (1972) 126 CLR 611
Jupe v Pratt (1837) 1 Web Pat Cas 145 (Ex)

L

Lane Fox v Kensington and Knightsbridge Electric Lighting Company (1892) 3 Ch 424
Le Roy v Tatham 55 US 156 (1852)
Liardet v Johnson (1778) 1 Carp Pat Cas 35 (NP)
Losh v Hague (1838) 1 Web Pat 202 (NP)

M

Mackay Radio v RCA 306 US 86 (1939)
Maeder v Busch (1938) 59 CLR 684
Merrill Lynch's Application [1988] RPC 1
Merrill Lynch's Application [1989] RPC 561 at 569

Microsoft Corporation v AT&T Corporation (Oral Argument) 05-1056, 21 February 2007

Millard v Commissioner of Patents (1918) 24 CLR 331

Morris v Bramson (1776) 1 Carp Pat Cas 30; 1 Abbot's PC 21 (KB)

Muntz v Foster (1843) 2 WPC 93

N

National Research Development Corporation v Commissioner of Patents (1959) 102 CLR 252

Nelson's Application [1980] RPC 173

Network Solutions, LLC [2011] APO 65 (19 August 2011)

New York Trust Co v Eisner 256 US 345 (1921)

NV Philips Gloeilampenfabrieken's Application (1966) 36 AOJP 2392

O

O'Reilly v Morse 56 US 62 (1852)

P

Palko v State of Connecticut 302 US 319 (1937)

Parker v Flook 437 US 584 (1978)

Patton (1942) 29 CCPA 982; 127 F.2d 324

Paturel v France App No 54968/00, 22 December 2005

Permutit Co v Borrowman (1926) 43 RPC 356

Peter Szabo and Associates Pty Ltd [2005] APO 24

Pitman's Application [1969] RPC 646

R

R v Wheeler (1819) 106 ER 392 (KB)

RCA Photophone Ltd v Gaumont-British Picture Corporation (1936) 53 RPC 167

Re Alsop's Patent (1907) 24 RPC 733

Re an Application for a Patent by Lenard (1954) 71 RPC 190

Re Application by Rantzen (1946) 64 RPC 63

Re C & W's Application for a Patent (1914) 31 RPC 235 (SG)

Bibliography

Re Cooper's Application for a Patent (1901) 19 RPC 53 (AG)
Re GEC's Application (1941) 60 RPC 7
Re Grant [2004] APO 11
Re IBM Corp's Application [1980] FSR 564
Re Lenard's Application (1954) 71 RPC 190
Re Standard Oil Development Co.'s Application (1951) 68 RPC 114
Reynolds v Herbert Smith & Co Ltd (1903) 20 RPC 123
Reynolds v Times Newspapers Ltd and Others (1999) 4 All ER 609
Roe v Wade 410 US 113 (1973)
Rogers v Commonwealth (1910) 10 CLR 701
Rubber-Tip Pencil Company v Howard 87 US 498 (1874)

S

SAS Institute Inc v World Programming Ltd [2010] EWHC 1829 (Ch)
Slee & Harris' Application [1966] FSR 51
State Street Bank & Trust Co v Signature Financial Group Inc 927 F. Supp. 502 (D Mass, 1996)
State Street Bank & Trust Co v Signature Financial Group Inc 149 F.3d 1368 (1998)
Symbian Ltd v Comptroller-General of Patents [2008] EWCA Civ 1066; [2009] RPC 1

T

T16/83 *Christian Franceries/Traffic Regulation* [1988] EPOR 65
T26/86 *Koch & Sterzel/X-Ray Apparatus* [1988] EPOR 72
T51/84 *Stockburger/Coded Distinctive Mark* [1986] EPOR 229
T115/85 *IBM/Computer-Related Invention* [1990] EPOR 107
T208/84 *Vicom/Computer-Related Invention* [1987] EPOR 74
T258/03 *Hitachi/Auction Method* [2004] EPOR 55
T366/87 *Sternheimer/Harmonic Vibrations* [1989] EPOR 131
T424/03 *Microsoft/Clipboard formats I* [2006] EPOR 39
T931/95 *PBS Partnership/Controlling pension benefits systems* [2002] EPOR 522
T935/97 *IBM's Application* [1999] RPC 861
T1173/97 *IBM/Computer program product* [1999] OJ EPO 609

T1227/05 *Circuit Simulation/Infineon Technologies* [2007] OJ EPO 574

Tilghman v Proctor 102 US 707 (1880)

Turner v Winter (1787) 19 Eng Rep 1276

U

Universal City Studios Inc v Reimerdes 82 F. Supp.2d 211 (SDNY, 2000)

V

Virginia-Carolina Chemical Corp's Application [1958] RPC 35

W

Whelan Associates v Jaslow Dental Lab Inc 797 F.2d 1222 (1986)

Williams Electronics v Artic International 685 F.2d 870 (3rd Circuit, 1986)

Welcome Real-Time SA v Catuity Inc and Ors (2001) 113 FCR 110

Y

Young v Rosenthal (1884) 1 RPC 29

Legislation

Charter of Human Rights and Responsibilities Act 2006 (Vic)

Copyright Act 17 USC (1976)

Copyright Act 1968 (Cth)

Patent Act of 1793, Ch. 11, 1 Stat. 318-323 (US)

Patents Act 1977 (UK)

Patents Act 1990 (Cth)

Patents Act 35 USC (1952)

Patents, Designs & Trade Marks Act 1883 (UK)

Statute of Monopolies 1623, 21 Jac 1, c 3 (UK)

Bibliography

Treaties

Convention on the Grant of European Patents, opened for signature 5 October 1973, 13 ILM 268 (entered into force 7 October 1973)

Agreement on Trade-Related Aspects of Intellectual Property Rights, opened for signature 15 April 1994, 1869 UNTS 299, 33 ILM 1197 (entered into force 1 January 1995)

International Covenant on Civil and Political Rights, opened for signature 16 December 1966, 999 UNTS 171 (entered into force 23 March 1976)

Patent Co-operation Treaty, [1980] ATS 6 (entered into force on 24 January 1978)

Universal Declaration of Human Rights, GA Res 217A (III), UN GAOR, 3rd sess, 183rd plen mtg, UN Doc A/810 (10 December 1948)

Other

"20th Century Mathematics" *The Story of Mathematics*,
<<http://www.storyofmathematics.com/20th.html>> (18 November 2010)

"A reading list in the 'classics,' from ancient to modern"
<<http://triviumquadrivium.wordpress.com/2010/02/13/is-it-worth-reading-the-classics-if-so-what-should-i-read/>> (30 March 2011)

"Altair" <<http://www.cis.usouthal.edu/faculty/daigle/project1/1975malt.htm>> (10 March 2010)

"Definition of Art" <<http://www.visual-arts-cork.com/art-definition>> (25 March 2011)

"Formalism" on *Bookrags.com* <<http://www.bookrags.com/research/formalism-wom/>> (14 July 2007)

"Hardware" in *The New Penguin Dictionary of Science* (Penguin, 2004)

"Hardware" on *Wiktionary* <<http://en.wiktionary.org/wiki/hardware>> (6 November 2008).

"lemma", *Princeton WordNet* <<http://wordnetweb.princeton.edu/perl/webwn?s=lemma>> (1 March 2011)

"mathematics", *Princeton WordNet* <<http://wordnetweb.princeton.edu/perl/webwn?s=mathematics>> (1 March 2011)

"Open Document Format for Office Applications (OpenDocument) v1.0" ISO/IEC 26300:2006
<http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=43485> (3 October 2008)

Patent Absurdity: How Software Patents Broke the System (Directed by Luca Lucarini, 2010)
<<http://patentabsurdity.com/>> (27 August 2011)

"Pop Quiz: What was the first personal computer?" *Blinkenlights Archaeological Institute*
<<http://www.blinkenlights.com/pc.shtml>> (10 March 2010)

The Macquarie Dictionary, (4th ed, The Macquarie Library Pty Ltd, 2003)

“The Software Industry in the 1950s” <http://www.softwarehistory.org/history/d_50s.html> (9 March 2010)

“The Software Industry in the 1960s” <http://www.softwarehistory.org/history/d_60s.html> (9 March 2010)

A

American Association for the Advancement of Science, “The Nature of Mathematics” on *Benchmarks Online* <<http://www.project2061.org/publications/bsl/online/ch9/ch9.htm>> (accessed 7 September 2011)

Apple Inc, “Introduction to Apple Human Interface Guidelines” *Apple Developer Connection* (2008)
<http://developer.apple.com/documentation/UserExperience/Conceptual/AppleHIGuidelines/XHIGIntro/chapter_1_section_1.html> (2 October 2008)

Arvind Asanovic, “Early Developments: From the Difference Engine to IBM 701” (Lecture Slides, MIT, 2005)
<http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-823-computer-system-architecture-fall-2005/lecture-notes/l01_earlydev.pdf> (11 September 2011)

Australian Bureau of Statistics, “Newsletters - Bulletin No. 12, June 2005”
<<http://bit.ly/qkgWOY>> (19 April 2007)

John Ayto, *Word Origins: the hidden histories of English words from A to Z* (A & C Black, 2006)

B

Robert Barr, Statement to the Joint Federal Trade Commission and Department of Justice Hearings on Economic Perspectives on Intellectual Property, Competition and Innovation, 28 February 2002, <<http://www.ftc.gov/opp/intellect/barrrobert.doc>> (5 September 2011)

Juan Batiz-Benet, Xuwen Cao and Yin Yin Wu, “CERN’s Large Hadron Collider”
<<https://sites.google.com/site/multinationalsoftwares/multinationalprojects/cern-lhc>> (4 September 2011)

Mary Bellis, “Inventors of the Modern Computer: Intel 4004 – The World’s First Single Chip Microprocessor” *About.com*
<<http://inventors.about.com/od/mstartinventions/a/microprocessor.htm>> (10 March 2010)

Mary Bellis, “The History of the Integrated Circuit aka Microchip” *About.com*
<http://inventors.about.com/od/istartinventions/a/integrated_circuit.htm> (10 March 2010)

Bibliography

Mary Bellis, "The History of the Transistor – John Bardeen, Walter Brattain and William Shockley" *About.com*

<<http://inventors.about.com/library/weekly/aa061698.htm>> (10 March 2010)

BPatG Fehlersuche 2000-07-28: Patentansprüche auf "Computerprogrammprodukt" etc unzulässig, BPatG /17W(pat)69/98, <<http://swpat.ffii.org/analysis/trips/index.en.html>> (11 October 2006)

Courtenay Brinckerhoff, "USPTO Backlog Update: One Step Forward, One Step Back" on *PharmaPatents*

<<http://www.pharmapatentsblog.com/patent-office-practice/uspto-backlog-update-one-step-forward-one-step-back/>> (16 August 2011)

John Brockman, "What Is Mathematics? A Talk With Reuben Hersh" *Edge* 5, 10 February 1997,

<<http://edge.org/documents/archive/edge5.html>> (6 December 2010)

Koni Buhner, "From Craft to Science: Searching for First Principles of Software Development" *The Rational Edge* (2000)

<<http://www.ibm.com/developerworks/rational/library/content/RationalEdge/archives/dec00.html>> (7 June 2007)

Business Software Alliance, "Software Industry Facts and Figures" <<http://bit.ly/nq8jKP>> (3 September 2011)

C

Piers Cawley, "Code is data, and it always has been" on *Just a Summary*

<<http://www.bofh.org.uk/2008/04/07/code-is-data-and-it-always-has-been>> (1 September 2010)

Richard Y Chappell, "On Liberty" on *Philosophy, et cetera* (2004)

<<http://www.philosophyetc.net/2004/06/on-liberty.html>> (26 June 2008)

Roger Clarke, "The Internet in Australia"

<<http://www.anu.edu.au/people/Roger.Clarke/II/OzI04.html>> (19 April 2007)

Tal Cohen, "An Interview with Douglas Hofstadter," 11 June 2008,

<http://tal.forum2.org/hofstadter_interview> (15 June 2008)

Computer History Museum, "Timeline of Computing"

<<http://www.computerhistory.org/timeline/?category=cmptr>> (9 March 2010)

B Jack Copeland, "The Church-Turing Thesis" on *AlanTuring.net* (2000)

<http://www.alanturing.net/turing_archive/pages/Reference%20Articles/The%20Turing-Church%20Thesis.html> (22 May 2007)

Dennis Crouch, "Bilski: Full CAFC to Reexamine the Scope of Subject Matter Patentability" on *Patently-O* (2008) <<http://www.patentlyo.com/patent/2008/02/bilski-ful-caf.html>> (accessed 18 September 2008)

D

Joseph W Dauben, "Galileo and Perspective: The Art of Renaissance Science" *American Physics Society* <<http://www.aps.org/publications/apsnews/200201/backpage.cfm>> (accessed 21 December 2007)

DataMonitor, "Software: Global Industry Guide 2010" 31 January 2011
<http://www.datamonitor.com/store/Product/software_global_industry_guide_2010?productid=4F026C5C-EBCC-4193-AD27-77260196E7F5> (3 September 2011)

Keith Devlin, "The mathematics of human thought" in *Devlin's Angle*, Mathematical Association of America, January 2004 <http://www.maa.org/devlin/devlin_01_04.html> (22 July 2011)

E

Encyclopaedia Britannica, "Calculating Clock" *Encyclopaedia Britannica Online*,
<<http://www.britannica.com/EBchecked/topic-art/725523/19230/The-Calculating-Clock-A-reproduction-of-Wilhelm-Schickards-Calculating-Clock>> (19 November 2009)

Encyclopaedia Britannica, "formal system" *Encyclopaedia Britannica Online*
<<http://www.britannica.com/EBchecked/topic/213751/formal-system>> (28 June 2011)

Encyclopaedia Britannica, "Parabolic satellite dish antenna" *Encyclopaedia Britannica Online*
<<http://www.britannica.com/EBchecked/media/67385>> (29 June 2011)

Encyclopaedia Britannica, "the arts" *Encyclopaedia Britannica Online*
<<http://www.britannica.com/EBchecked/topic/36405/the-arts>> (10 August 2011)

Encyclopaedia Britannica, "Satellite communication: satellite dish antenna" *Encyclopaedia Britannica Online* <<http://www.britannica.com/eb/art-67385>> (27 December 2007)

European Commission, "Proposal (EC) 2002/92 for a Directive of the European Parliament and of the Council on the patentability of computer-implemented inventions" [2002] OJ C151/129

F

FindLaw Australia, "Businesses Innovating, But Few Protecting IP" 22 February 2005
<<http://www.findlaw.com.au/news/default.asp?task=read&id=22785&site=LE>> (28 June 2011)

Foundation for a Free Information Infrastructure, "Alliance of 2,000,000 SMEs against Software Patents and EU Directive" <<http://eupat.ffii.org/papers/eubsa-swp0202/ceapme0309/>> (18 September 2010)

Foundation for a Free Information Infrastructure, "Quotations on Software Patents,"
<<http://eupat.ffii.org/vreji/quotes/index.en.html>> (18 September 2010)

Foundation for a Free Information Infrastructure, "Regulation about the invention concept of the European patent system and its interpretation with special regard to programs for computers", <<http://swpat.ffii.org/stidi/javni/index.en.html>> (12 October 2006)

Bibliography

Philip L Frana, "An Interview with Donald E Knuth", *University of Minnesota Digital Conservancy*, 8 November 2001 <<http://conservancy.umn.edu/bitstream/107413/1/oh332dk.pdf>> (1 August 2011).

Cameron Freer, "What is vdash?" <<http://www.vdash.org/intro/>> (11 September 2011)

G

Bill Gates, "Challenges and Strategy" (Microsoft Internal Memo, 1991)
<<http://bralyn.net/etext/literature/bill.gates/challenges-strategy.txt>> (20 April 2007)

Fernando Gouvea, "Book Review: What is Mathematics Really? by Reuben Hersh," *MAA Online*, <<http://test.maa.org/reviews/whatis.html>> (9 September 2011)

Paul Graham "Great Hackers" <<http://www.paulgraham.com/gh.html>> (16 June 2011)

Paul Graham "Hackers and Painters" <<http://www.paulgraham.com/hp.html>> (1 August 2011)

H

John R Harrison "Chapter 3 - Lambda calculus as a programming language" *Introduction to Functional Programming* (Lecture Notes, Cambridge University, 1997)
<<http://www.cl.cam.ac.uk/teaching/Lectures/funprog-jrh-1996/>> (5 November 2008)

Andrew Hunt and David Thomas, "The Art in Computer Programming" September 2001
<<http://media.pragprog.com/articles/other-published-articles/ArtInProgramming.pdf>> (1 August 2011)

I

Internet Engineering Task Force, "Request for Comments 1122: Requirements for Internet Hosts"
<<http://tools.ietf.org/html/rfc1122>> (8 October 2008)

IP Australia, "Glossary" <<http://www.ipaustralia.gov.au/resources/glossary.shtml>> (25 February 2010)

J

Jenkins & Co, "Computer Program Inventions Before the German Supreme Court"
<<http://www.jenkins.eu/articles-patents/computer-program-inventions—germany.asp>> (17 August 2011)

K

Ben Klemens, "The Supreme Court's Patent Trilogy: An Analysis" The Brookings Institution (2006) <<http://www.brook.edu/views/op-ed/20060530klemens.htm>> (accessed 27 April 2007)

John Kopplin, "An Illustrated History of Computers" *ComputerScienceLab.com*, 2002
<<http://www.computersciencelab.com/ComputerHistory/History.htm>> (10 March 2010)

Greg Kroah-Hartman, Jonathan Corbet and Amanda McPherson, "Linux Kernel Development (April 2008)" *The Linux Foundation* (2008)
<<http://www.linuxfoundation.org/publications/linuxkerneldevelopment.php>> (accessed 28 August 2008)

L

Michael Lahanas, "Irrational numbers" <<http://www.mlahanas.de/Greeks/Irrational.htm>> (28 December 2007)

Jean Lassègue, "Doing Justice to the Imitation Game; a Farewell to Formalism," December 2003,
<http://formes-symboliques.org/article.php3?id_article=75> (10 September 2011).

League for Programming Freedom, "Against Software Patents" 28 February 1991
<<http://www.progfree.org/Patents/against-software-patents.html>> (5 September 2011)

Angel Leon, "How many lines of code does it take to create the Android OS?" on *Gubatron.com*
<<http://www.gubatron.com/blog/2010/05/23/how-many-lines-of-code-does-it-take-to-create-the-android-os/>> (4 September 2011)

Peter Lyman and Hal R Varian, "How Much Info? Internet" (School of Information Management and Systems, University of California, 2000)
<<http://www2.sims.berkeley.edu/research/projects/how-much-info/internet.html>> (7 September 2011)

M

Richard Mansfield, "Real and Complex Numbers" in *Logic: The Bridge to Higher Math*
<<http://www.math.psu.edu/melvin/logic/node8.html>> (7 September 2011)

Bob Marley, "No Woman, No Cry", *Live!* (1975)

John A Masters, "Imaginary Numbers Really Exist! (I'm Really, Really Serious!)" on *Least Significant Bits* <<http://jamspeaks.blogspot.com/2007/09/imaginary-numbers-really-exist-im.html>> (7 September 2011)

Matt McCormick, "Immanuel Kant: Metaphysics" *Internet Encyclopedia of Philosophy*, 30 June 2005 <<http://www.iep.utm.edu/kantmeta/>> (7 December 2010)

Daniel W McDonald, Robert A Kalinsky and William D Schulz, "Software Patent Litigation" (Intellectual Property Litigation Committee Roundtable Discussion, ABA Section of Litigation,

Bibliography

April 2006)

<<http://euro.econ.cmu.edu/program/law/08-732/Patents/SoftwarePatentLitigation.pdf>> (20 August 2010)

Ross McFarlane and Stephen O'Brien, "Physical effect of business methods" *Phillips, Ormonde & Fitzpatrick*, 14 August 2006 <<http://www.pof.com.au/Physicaleffectofbusinessmethods>> (29 June 2011)

Robert McMillan, "Torvalds joins in anti-patent attack" *Techworld*, 2 February 2005, <<http://news.techworld.com/applications/3059/torvalds-joins-in-anti-patent-attack/>> (18 September 2010)

Microsoft Legal and Corporate Affairs, "Patent Cooperation Agreement - Microsoft and Novell Interoperability Collaboration" on *Microsoft.com*, 2 November 2006 <http://www.microsoft.com/about/legal/en/us/IntellectualProperty/IPLicensing/customer covenant/msnovellcollab/patent_agreement.aspx> (4 September 2011)

N

Brian Napper, "The Manchester Small Scale Experimental Machine – 'The Baby'" <<http://www.computer50.org/mark1/new.baby.html>> (14 September 2011)

Karim Nice, "How Car Computers Work" on *HowStuffWorks.com* <<http://auto.howstuffworks.com/under-the-hood/trends-innovations/car-computer.htm>> (4 September 2011)

O

Larry O'Brien, "How Many Lines of Code in Windows?" on Larry O'Brien, *Knowing.NET* (6 December 2005) <<http://www.knowing.net/index.php/2005/12/06/how-many-lines-of-code-in-windows/>> (28 June 2011)

Tim O'Reilly, "Ask Tim: Open Source, Patents & O'Reilly" *O'Reilly Network*, 28 February 2000 <http://oreilly.com/pub/a/oreilly/ask_tim/2000/amazon_patent.html> (18 September 2010)

Oyez, "Gottschalk v Benson" <http://www.oyez.org/cases/1970-1979/1972/1972_71_485/> (26 Feb 2008)

P

Ryan Paul, "EFF files brief in free speech case" on *Ars Technica* (17 April 2006) <<http://arstechnica.com/news.ars/post/20060417-6609.html>> (accessed 23 April 2007)

Bruce Perens, "Public Policy Area: Software Patents" <<http://perens.com/policy/software-patents/>> (18 September 2010)

Bruce Perens, "The Monster Arrives: Software Patent Lawsuits Against Open Source Developers" <<http://technocrat.net/d/2006/6/30/5032/>> (18 September 2010)

Karlis Podnieks, "Platonism, Intuition and the Nature of Mathematics" in *Around Goedel's theorem* <<http://www.ltn.lv/~podnieks/gt1.html>> (accessed 29 October 2007)

Letter from Professor Alain Pompidou to Lord Justice Jacob, 22 February 2007
<<http://www.ipo.gov.uk/p-pn-subjectmatter-letter.htm>> (25 May 2010)

R

Dave Raggett "Introduction to CSS" <<http://www.w3.org/MarkUp/Guide/Style>> (4 November 2008)

Todd Rowland, "Church-Turing Thesis" *Wolfram MathWorld* (2002)
<<http://mathworld.wolfram.com/Church-TuringThesis.html>> (accessed 23 April 2007)

S

Gerhard Schmidt, "Mult8.asm" on *Tutorial for learning assembly language for the AVR-Single-Chip-Processors* <http://www.avr-asm-tutorial.net/avr_en/calc/MULT8E.html> (24 July 2008)

Claude E Shannon, *A symbolic analysis of relay and switching circuits* (Massachusetts Institute of Technology, Dept. of Electrical Engineering, 1940)

Jeff Silvis and Mark Kowitt, "The Four Forces of Nature," *Ask An Astrophysicist*,
<http://imagine.gsfc.nasa.gov/docs/ask_astro/answers/980127c.html> (9 May 2011)

SpringerLink, "Computable function" *Encyclopaedia of Mathematics* (2001)
<<http://eom.springer.de/c/c024330.htm>> (31 July 2008)

SpringerLink, "Formalism" *Encyclopaedia of Mathematics* (2001)
<<http://eom.springer.de/f/f040910.htm>> (17 May 2007)

SpringerLink, "Gödel incompleteness theorem" *Encyclopaedia of Mathematics*
<<http://eom.springer.de/g/g044530.htm>> (17 May 2007)

Richard Stallman, "The Danger of Software Patents," Transcript of Speech Given at Cambridge University, March 2002, <<http://www.cl.cam.ac.uk/~mgk25/stallman-patents.html>> (27 August 2011)

Peter Suber, "Formal Systems and Machines: An Isomorphism"
<<http://www.earlham.edu/~peters/courses/logsys/machines.htm>> (accessed 27 June 2011)

Peter Suber, "Legal Reasoning" <<http://www.earlham.edu/~peters/writing/leglreas.htm>> (accessed 6 January 2008)

Bibliography

T

Thomas Taylor, "Obfuscation, Weird Languages and Code Aesthetics"

<<http://www.obfuscators.org/2008/04/obfuscation-weird-languages-and-code.html>> (1 August 2011)

Linus Torvalds and Alan Cox, "Open Letter on Software Patents from Linux Developers"

21 September, 2003 <http://www.effi.org/patentit/patents_torvalds_cox.html> (5 September 2011)

United States Department of Energy, "About the Human Genome Project"

<http://www.ornl.gov/sci/techresources/Human_Genome/about.shtml> (4 September 2011)

U

United States Patent and Trademark Office, "Public Hearing on Use of the Patent System to Protect Software-Related Inventions", Transcript of Proceedings, San Jose California , 26-27 January 1994

<<http://euro.ecom.cmu.edu/program/law/08-732/Patents/SoftwarePatentHearings.pdf>> (20 August 2010)

United States Patent and Trademark Office, "Fiscal Year 2009: Patent Applications Pending Prior to Allowance"

<http://www.uspto.gov/web/offices/com/annual/2009/oai_05_wlt_03.html> (12 July 2010)

United States Patent and Trademark Office "Backlog Stimulus Reduction Plan"

<http://www.uspto.gov/patents/init_events/PatentStimulusPlan.jsp> (17 August 2011)

United States Patent and Trademark Office, "July 2011 Patents Data, at a Glance"

<<http://www.uspto.gov/dashboards/patents/main.dashxml>> (15 August 2011)

W

Larry Wall, "Perl, the first postmodern computer language"

<<http://www.perl.com/pub/1999/03/pm.html>> (7 July 2011)

Stephen White, "A Brief History of Computing - Complete Timeline" (2005)

<<http://trillian.randomstuff.org.uk/~stephen/history/timeline.html>> (accessed 19 April 2007)

David Wheeler, "Counting Source Lines of Code (SLOC)" <<http://www.dwheeler.com/sloc/>> (accessed 28 August 2008)

Wikipedia, "Altair 8800" <http://en.wikipedia.org/wiki/Altair_8800> (10 March 2010)

Wikipedia, "Assembly language" <http://en.wikipedia.org/wiki/Assembly_language> (18 August 2008)

Wikipedia, "Church-Turing thesis" <http://en.wikipedia.org/wiki/Church-Turing_thesis> (22 May 2007)

Wikipedia, "Code Forking" <http://en.wikipedia.org/wiki/Code_forking> (18 September 2010)

Wikipedia, "Computer simulation" <http://en.wikipedia.org/wiki/Computer_simulation> (4 November 2008)

Wikipedia, "CSIRAC" <<http://en.wikipedia.org/wiki/CSIRAC>> (7 September 2008)

Wikipedia, "Design pattern (computer science)"
<[http://en.wikipedia.org/wiki/Design_pattern_\(computer_science\)](http://en.wikipedia.org/wiki/Design_pattern_(computer_science))> (29 July 2008)

Wikipedia, "Fields Medal" <http://en.wikipedia.org/wiki/Fields_Medal> (25 June 2011)

Wikipedia, "Influence of the IBM-PC on the personal computer market"
<http://en.wikipedia.org/wiki/Influence_of_the_IBM-PC_on_the_personal_computer_market> (30 September 2010)

Wikipedia, "Interactive theorem proving"
<http://en.wikipedia.org/wiki/Interactive_theorem_proving> (4 November 2008)

Wikipedia, "Kernel (computer science)"
<[http://en.wikipedia.org/wiki/Kernel_\(computer_science\)](http://en.wikipedia.org/wiki/Kernel_(computer_science))> (23 July 2008)

Wikipedia, "Large Hadron Collider" <http://en.wikipedia.org/wiki/Large_Hadron_Collider> (4 September 2011)

Wikipedia, "Lambda calculus" <http://en.wikipedia.org/wiki/Lambda_calculus> (5 November 2008)

Wikipedia, "Liberal arts" <http://en.wikipedia.org/wiki/Liberal_arts> (17 May 2007)

Wikipedia, "Mathematics" <<http://en.wikipedia.org/wiki/Mathematics>> (17 May 2007)

Wikipedia, "Moore's Law" <http://en.wikipedia.org/wiki/Moore%27s_law> (18 September 2010)

Wikipedia, "Mutually Assured Destruction,"
<http://en.wikipedia.org/wiki/Mutually_assured_destruction> (18 September 2010)

Wikipedia, "Obfuscated code" <http://en.wikipedia.org/wiki/Obfuscated_code> (11 September 2011)

Wikipedia, "Operating system" <http://en.wikipedia.org/wiki/Operating_system> (23 July 2008)

Wikipedia, "Parabola" <<http://en.wikipedia.org/wiki/Parabola>> (30 October 2007)

Wikipedia, "Patent Troll," <http://en.wikipedia.org/wiki/Patent_troll> (18 September 2010)

Wikipedia, "Philosophy of mathematics"
<http://en.wikipedia.org/wiki/Philosophy_of_Mathematics> (20 April 2007)

Wikipedia, "Pseudocode" <<http://en.wikipedia.org/wiki/Pseudocode>> (8 August 2011)

Wikipedia, "Standard" *Wikipedia.org* <<http://en.wikipedia.org/wiki/Standard>> (3 October 2008)

Bibliography

Wikipedia, "Stored Program"

<http://en.wikipedia.org/wiki/Stored_program> (7 September 2009)

Wikipedia, "TCP/IP" <<http://en.wikipedia.org/wiki/TCP/IP>> (8 October 2008)

Wikipedia, "Type theory" <http://en.wikipedia.org/wiki/Type_theory> (30 November 2010)

Otto Willmann, "The Seven Liberal Arts" in *The Catholic Encyclopedia*, Volume 1 (Robert Appleton Company, 1907) <<http://www.newadvent.org/cathen/01760a.htm>> (31 Mar 2011)

Don Woligroski, "The Basic Parts of a Graphics Card" *Tom's Hardware*, 24 July 2006

<<http://www.tomshardware.com/reviews/graphics-beginners,1288-2.html>> (15 October 2010)

World Intellectual Property Organisation, "Online Forum on Intellectual Property in the Information Society"

<<http://wipo.int/roller/comments/ipisforum/Weblog>> (accessed 24 April 2007)

Jon Wuebben, "The History of the Floppy and Hard Disk Drive" (20 March 2007)

<<http://www.patantconsult.com/articlesvault/Article/The-History-of-the-Floppy-and-Hard-Disk-Drive/11501>> (11 September 2011)

X

Xandros Inc, "Microsoft, Xandros Broad Collaboration Agreement Extends Bridge Between Commercial Open Source and Microsoft Software" on *xandros.com*, 4 June 2007

<http://www.xandros.com/news/press_releases/xandros_microsoft_collaborate.html> (4 September 2011)

Y

Lynn Yaris, "Machines and the Human Genome Project"

<www.lbl.gov/Science-Articles/Archive/human-genome-mapping-sequencing.html> (4 September 2011)